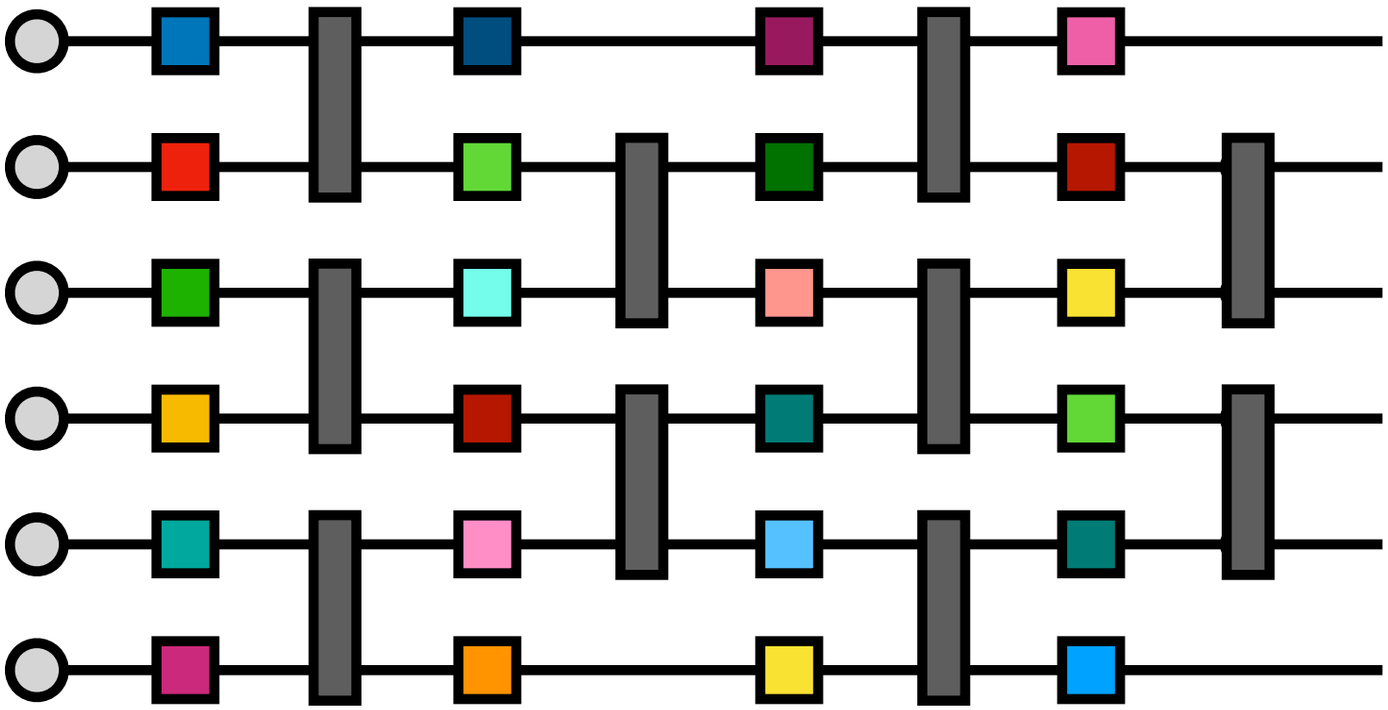
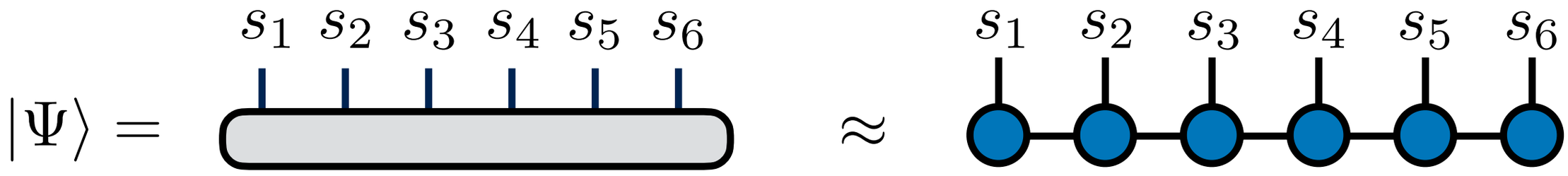
Introduction to Tensor Networks





E.M. Stoudenmire

August 2024 - U. Mass. Amherst

FLATIRON INSTITUTE
SIMONS FOUNDATION

Flatiron Institute

FLATIRON INSTITUTE

SIMONS FOUNDATION

The mission of the Flatiron Institute is to advance scientific research through computational methods, including data analysis, modeling and simulation.

CCA: Center for Computational Astrophysics

CCB: Center for Computational Biology

CCQ: Center for Computational Quantum Physics

CCM: Center for Computational Mathematics

CCN: Center for Computational Neuroscience

ICC: Initiative for Computational Catalysis

Who are we?

Karl Pierce

Flatiron Software Fellow

High-performance computing

"CP" tensor factorizations & applications

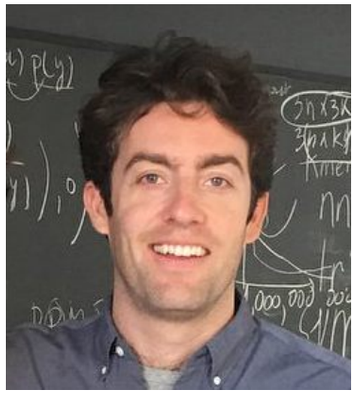
Miles Stoudenmire

Research Scientist

Tensor network algorithms

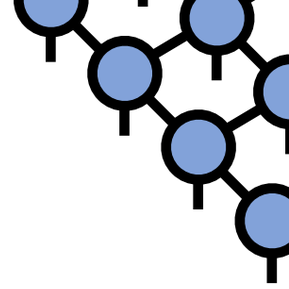
Applications to condensed-matter physics

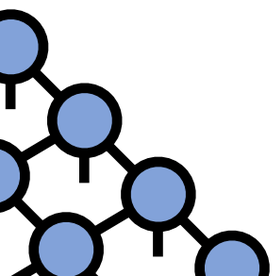
Broader applications (machine learning, classical physics)



Plan for Today

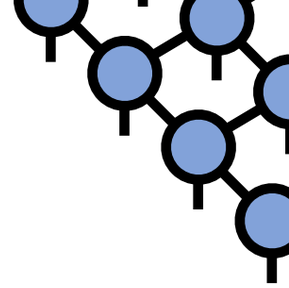
- Tensors and Tensor Networks
- Quantum Circuits as Tensor Networks
- Intro to the ITensor Software
- Algorithms for Tensor Networks (MPS and MPO)
- Using ITensor on GPU
- Future of Tensor Networks: Functions & Machine Learning

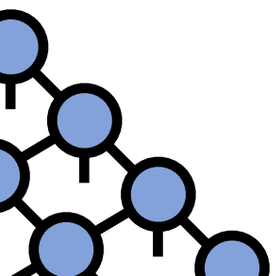




Plan for Today

- Tensors and Tensor Networks
- Quantum Circuits as Tensor Networks
- Intro to the ITensor Software
- Algorithms for Tensor Networks (MPS and MPO)
- Using ITensor on GPU
- Future of Tensor Networks: Functions & Machine Learning





Introduction to Tensors

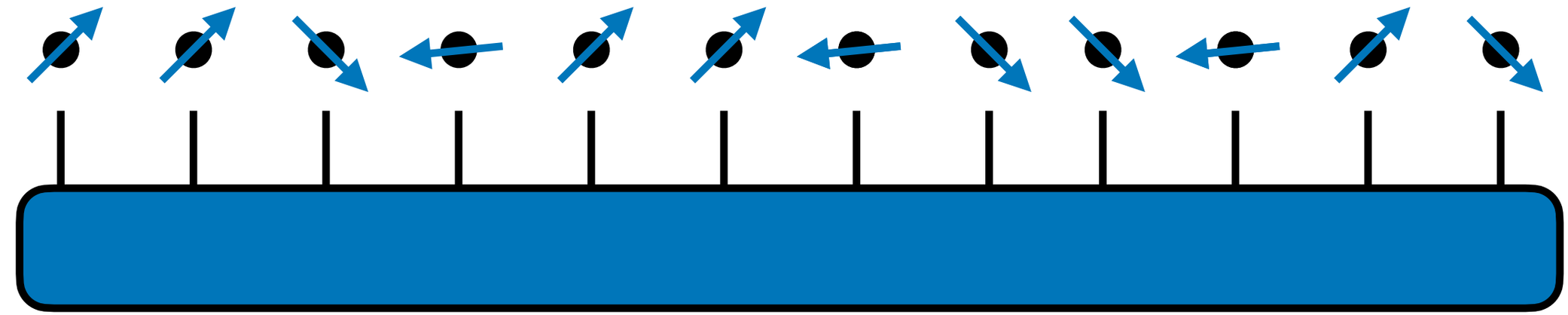
What is a tensor?

Why are they challenging?

Classical Methods for Quantum Systems

How hard is quantum mechanics?

Consider n quantum spins or qubits



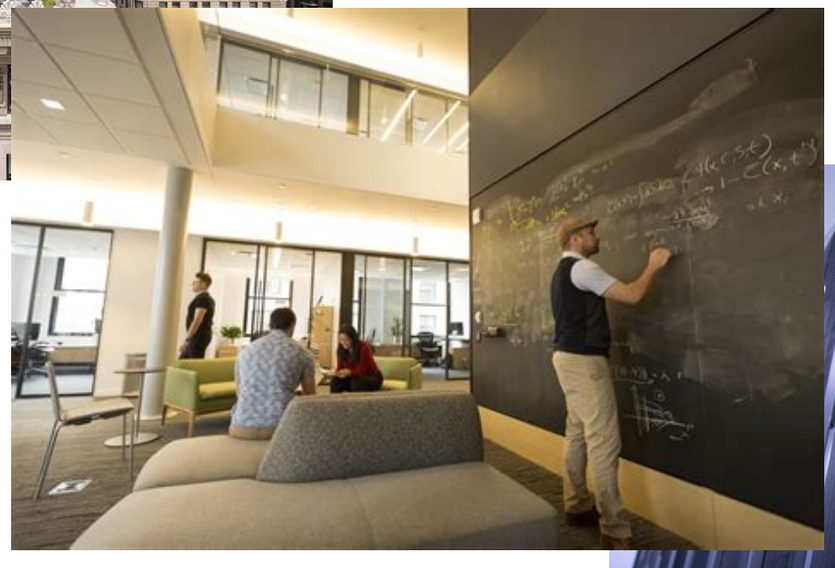
quantum wavefunction Ψ

2^n parameters inside

exponentially hard to store & manipulate

Classical Methods for Quantum Systems

What's going on at Flatiron Institute
Center for Computational Quantum Physics (CCQ)
in New York?



Classical Methods for Quantum Systems

Developing ways to break through the 2^n exponential quantum wall

quantum Monte Carlo

high-order perturbation theory

embedding (DMFT)

GW method

tensor networks

neural quantum states

density functional theory

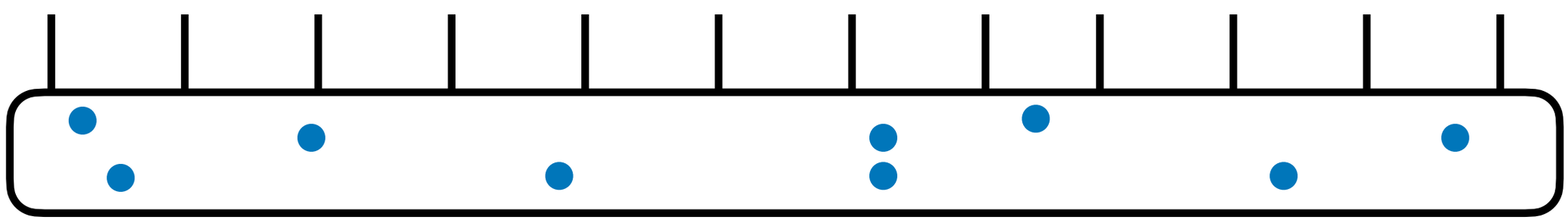
numerical renormalization group

Classical Methods for Quantum Systems

Developing ways to break through the 2^n exponential quantum wall

Quantum Monte Carlo 🎲

breaks exponential by sampling important configurations



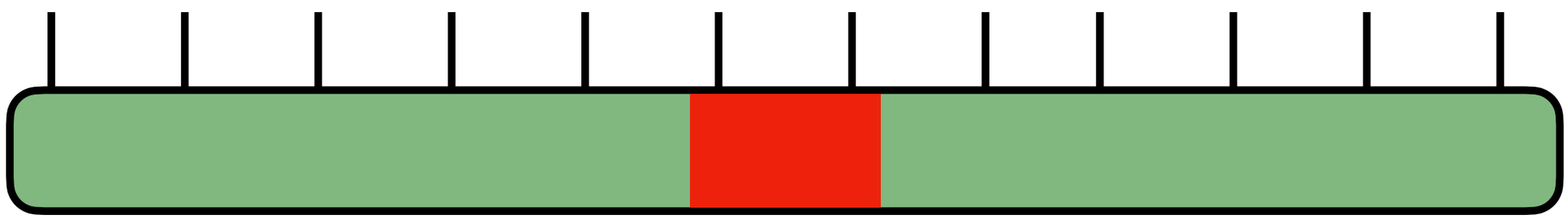
Classical Methods for Quantum Systems

Developing ways to break through the 2^n exponential quantum wall

Embedding / DMFT 🎯

treats small piece of system

inside solvable "bath" with mirrored properties



Classical Methods for Quantum Systems

Last but not least: **tensor networks**

Tensor networks:

- work directly with wavefunction
- use compression to store wavefunction
- closely mimic a quantum computer

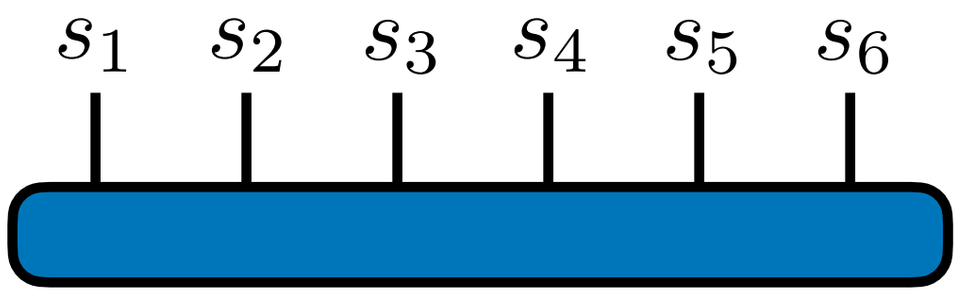
Classical Methods for Quantum Systems

General wavefunction of n qubits

$$|\Psi\rangle = \sum_{s_1 s_2 s_3 \cdots s_n} \Psi^{s_1 s_2 s_3 \cdots s_n} \left| s_1 s_2 s_3 \cdots s_n \right\rangle \qquad s_j \in 0, 1$$

Amplitudes form a big tensor!

$$\Psi^{s_1 s_2 s_3 s_4 s_5 s_6} =$$



Classical Methods for Quantum Systems

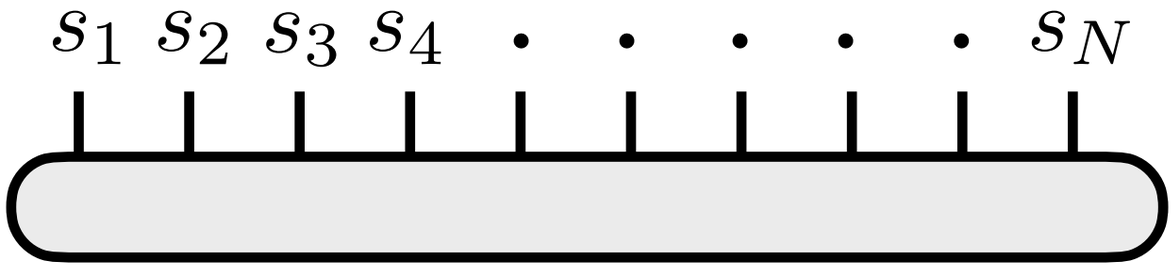
What is a tensor?

vector	$v = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$	$v_2 = 3$
matrix	$M = \begin{bmatrix} 5 & 7 \\ 8 & 9 \end{bmatrix}$	$M_{12} = 7$
order-3 tensor	$T = \begin{bmatrix} 3 \begin{bmatrix} 5 & 4 \end{bmatrix} 7 \\ 1 \begin{bmatrix} 3 & 2 \end{bmatrix} 5 \end{bmatrix}$	$T_{112} = 5$

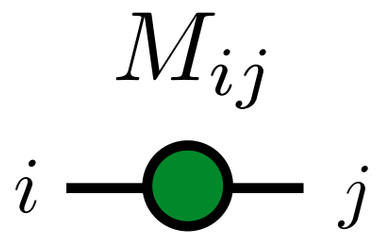
Classical Methods for Quantum Systems

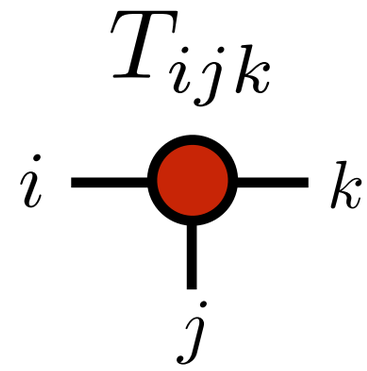
N-index tensor = shape with N lines

$$T^{s_1 s_2 s_3 \cdots s_N} =$$



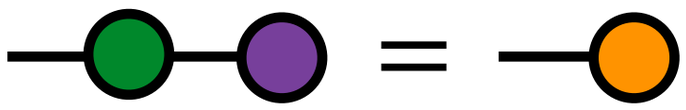
Low-order examples:



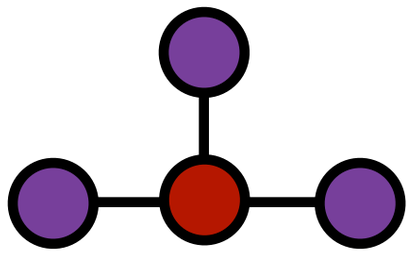


Tensors – Penrose Diagram Notation

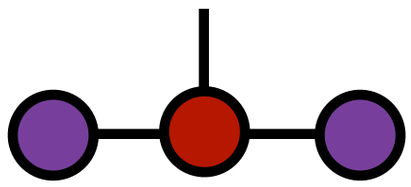
Joining wires means contraction:



$$\sum_j M_{ij} v_j = w_i$$



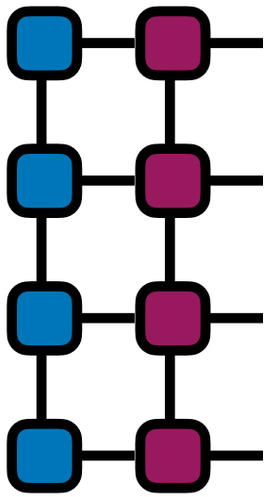
$$\sum_{i,j,k} T^{ijk} v_i v_j v_k$$



$$\sum_{i,k} T^{ijk} v_i v_k = z^j$$

Tensors – Penrose Diagram Notation

Much simpler than conventional expressions



$$= \sum_{\{\alpha\},\{\beta\},\{s\}} M^{s_1}_{\alpha_1} M^{s_2}_{\alpha_1,\alpha_2} M^{s_3}_{\alpha_2,\alpha_3} M^{s_4}_{\alpha_3}$$

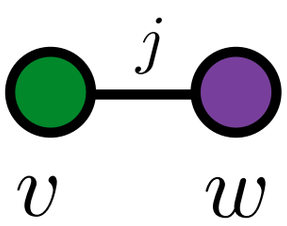
$$\cdot W^{s_1,t_1}_{\beta_1} W^{s_2,t_2}_{\beta_1,\beta_2} W^{s_3,t_3}_{\beta_2,\beta_3} W^{s_4,t_4}_{\beta_3}$$

Equally rigorous

When using circuit diagrams, you're using this notation!

Tensors – Penrose Diagram Notation

Test your knowledge – describe these operations
in your own words

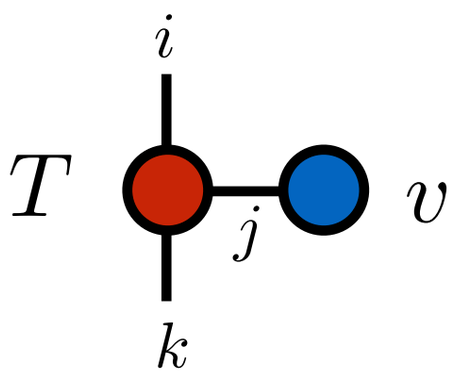


$$\longleftrightarrow \quad \sum_j v_j w_j$$

vector inner product

$$\longleftrightarrow \quad \sum_n M_{nn}$$

trace of a matrix



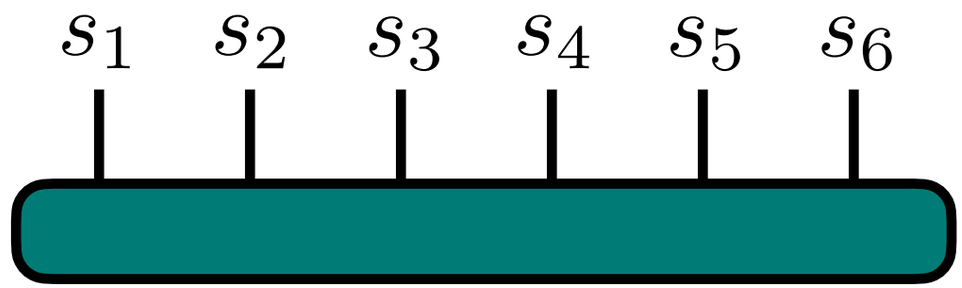
$$\longleftrightarrow \quad \sum_j T_{ijk} v_j = R_{ik}$$

contract order-3 tensor with vector

Tensors

Any function of discrete variables can be represented as a tensor

$$f(s_1, s_2, s_3, s_4, s_5, s_6) =$$

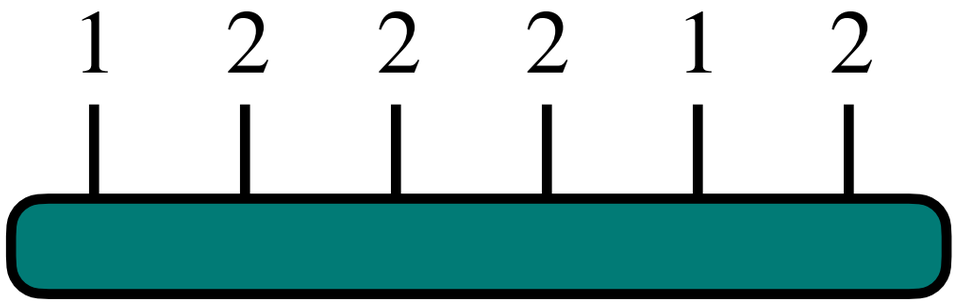


All values of f inside

Tensors

Any function of discrete variables can be represented as a tensor

$$f(1,2,2,2,1,2) \;=$$

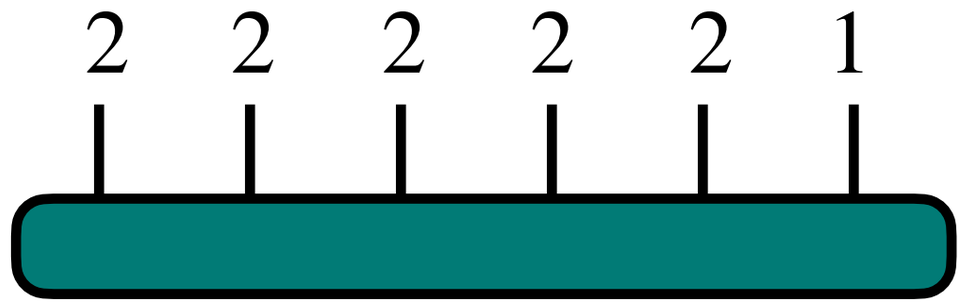


$$= \quad 1.2$$

Tensors

Any function of discrete variables can be represented as a tensor

$$f(2,2,2,2,2,1) = $$

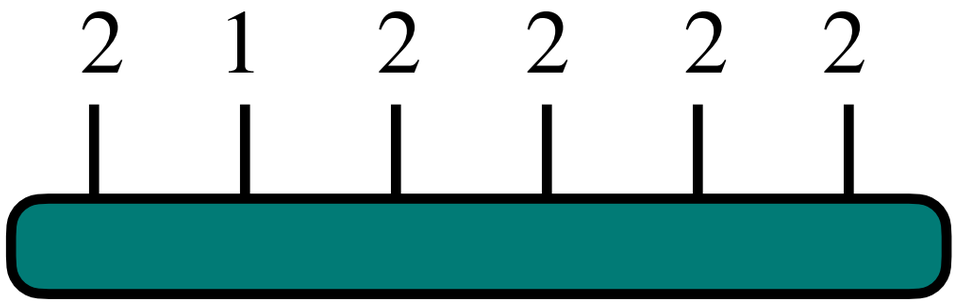


$$= \quad 0.3$$

Tensors

Any function of discrete variables can be represented as a tensor

$$f(2,1,2,2,2,2) =$$

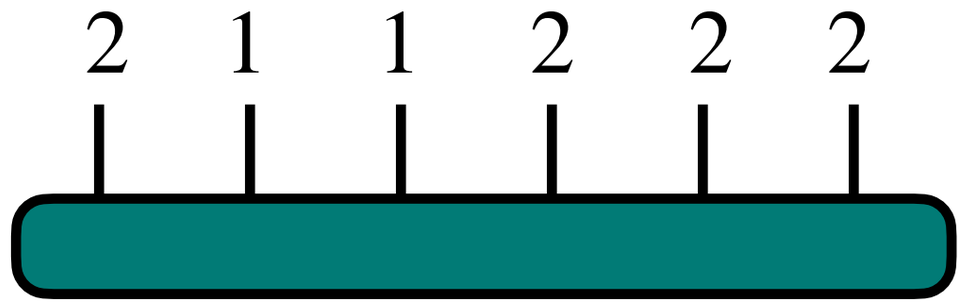


$$= -0.2$$

Tensors

Any function of discrete variables can be represented as a tensor

$$f(2,1,1,2,2,2) \;=\;$$

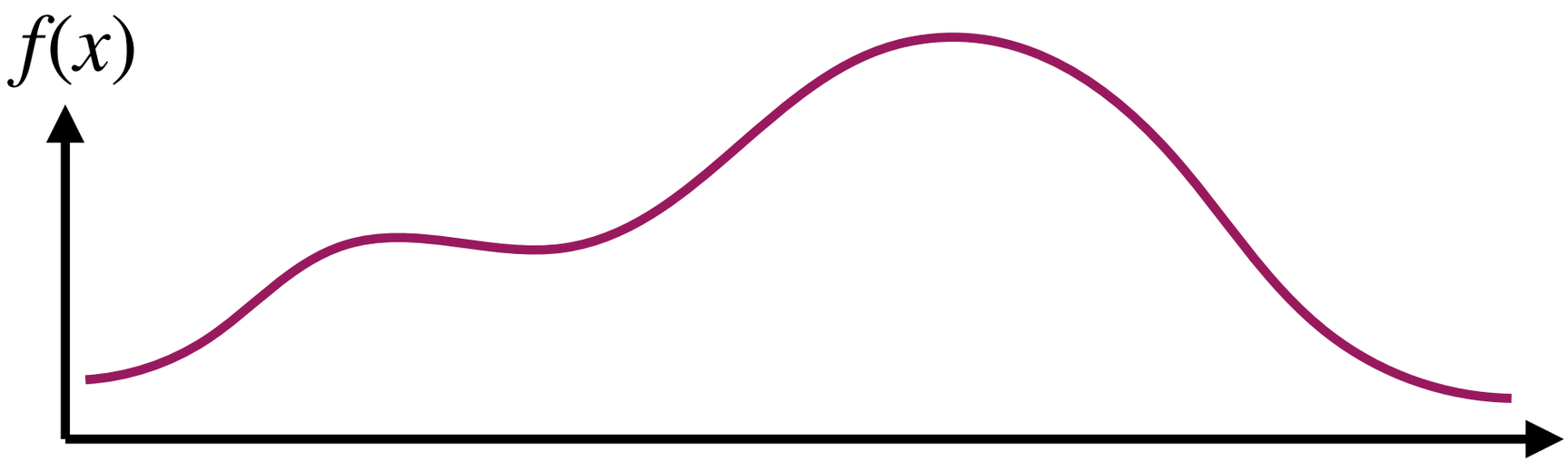


$$= \quad 2.7$$

Tensors

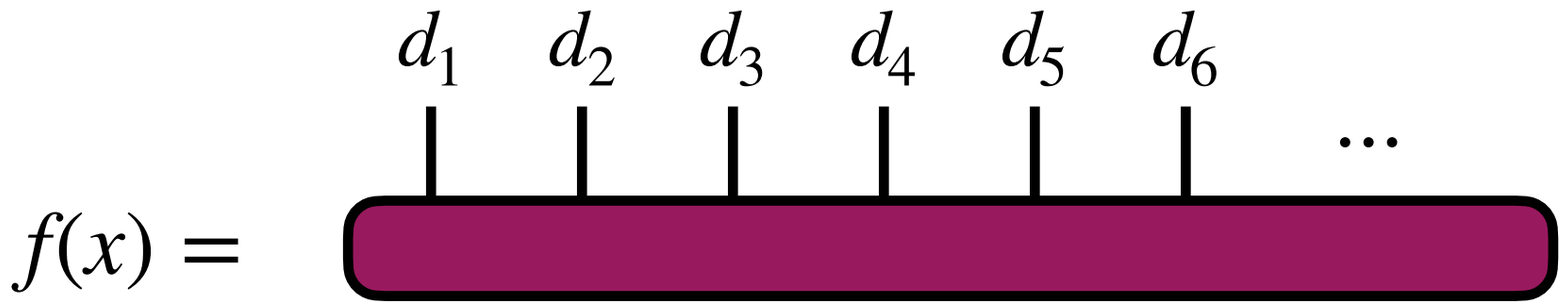
By a simple trick, can encode **continuous arguments** too

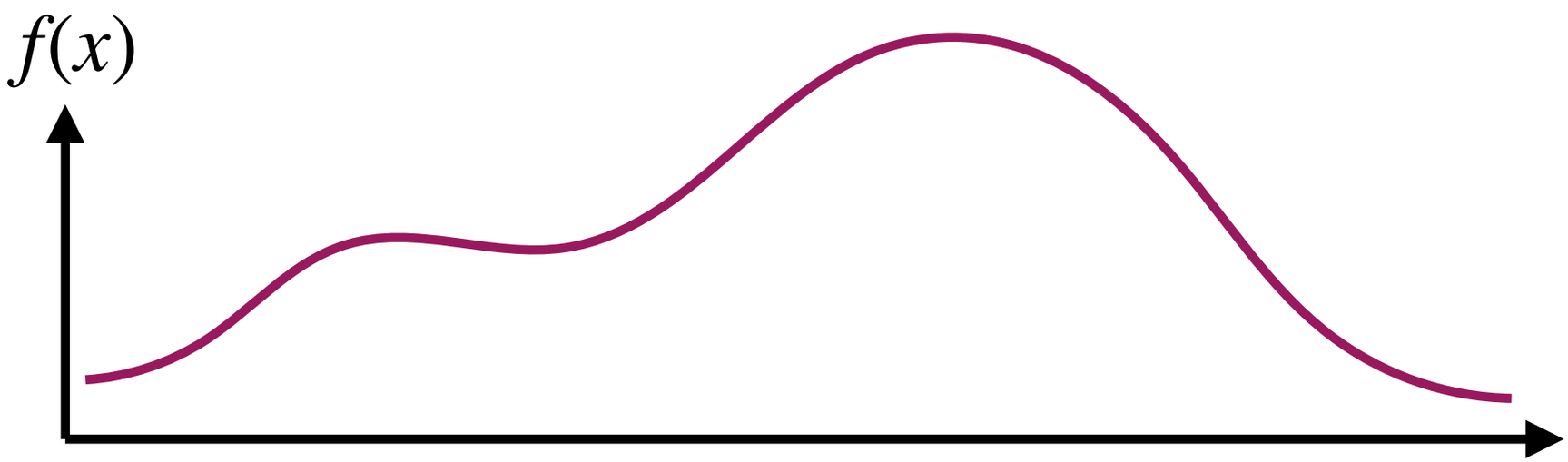
$$f(x) =$$



Tensors

By a simple trick, can encode
continuous arguments too

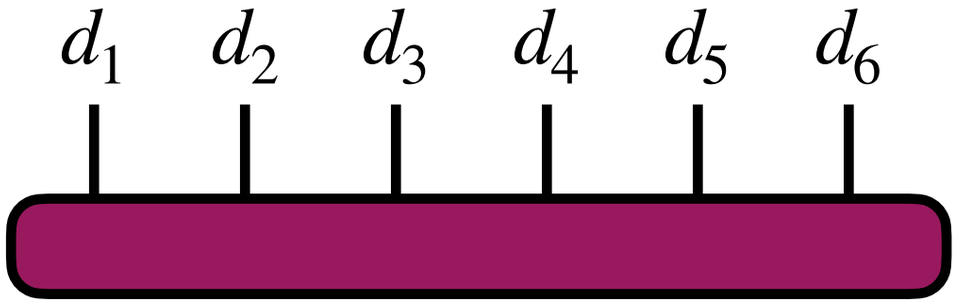


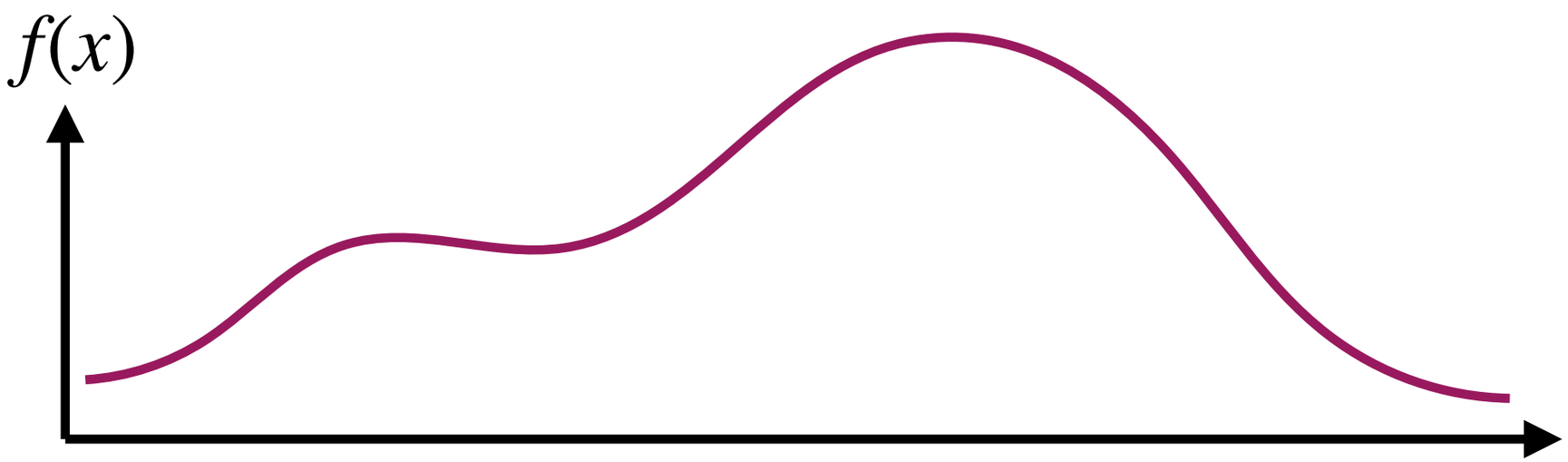


Tensors

By a simple trick, can encode
continuous arguments too

$$f(x) \approx f(0.d_1d_2d_3d_4d_5d_6) =$$

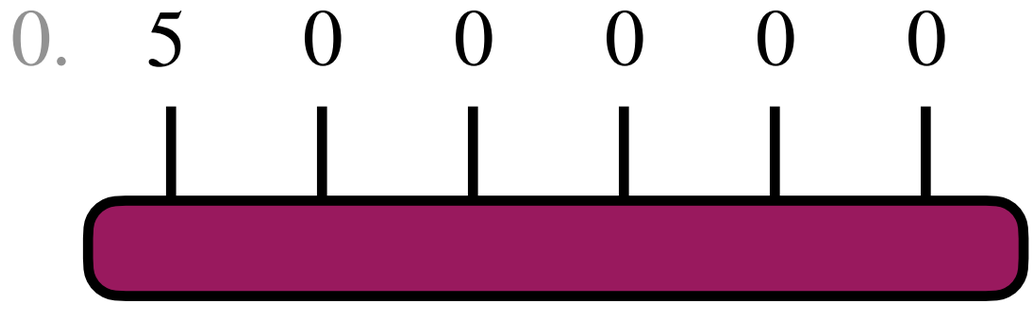


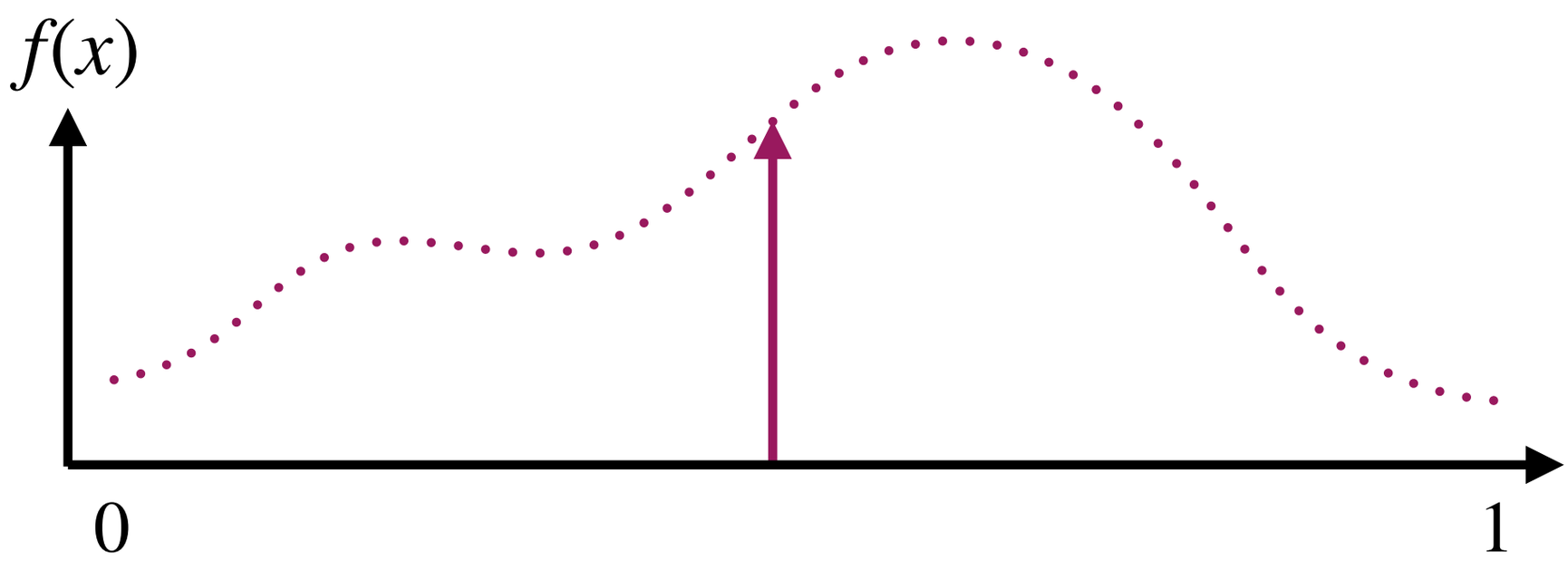


Why Tensors?

By a simple trick, can encode
continuous arguments too

$$f(x) \approx f(0.500000) =$$

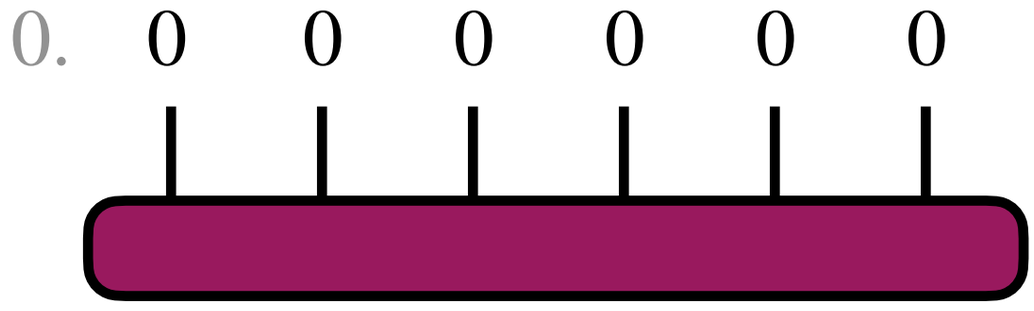


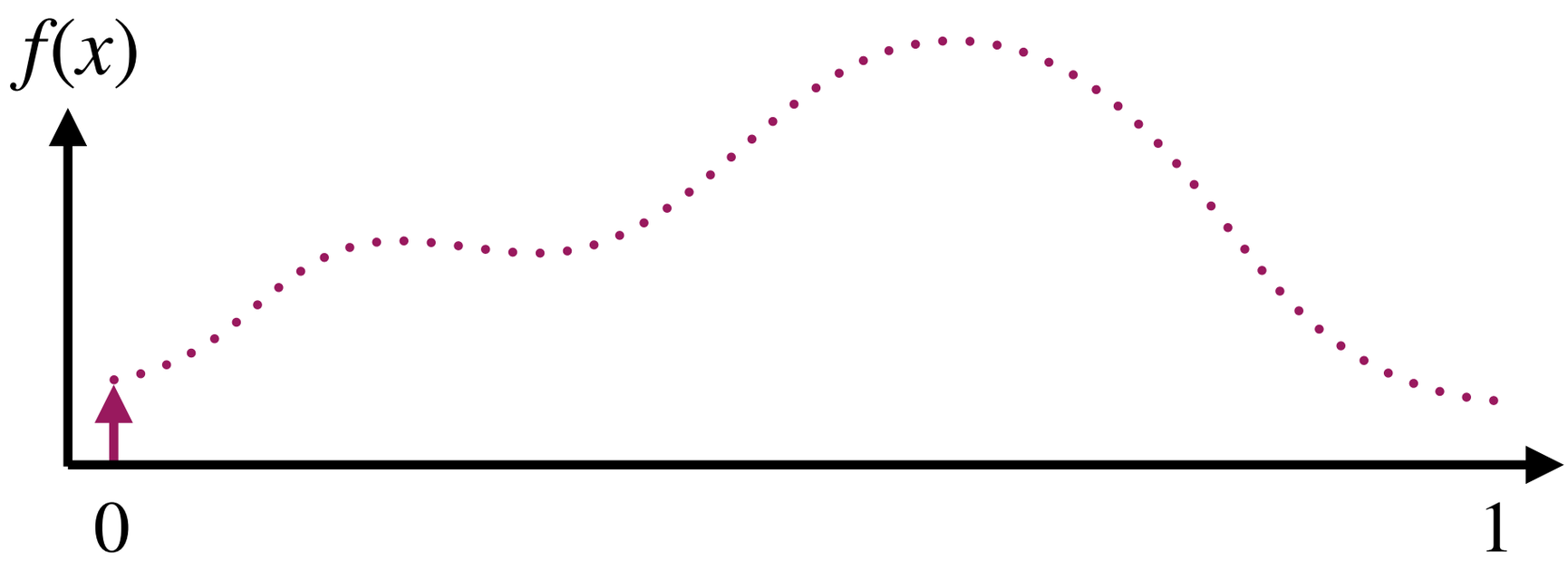


Why Tensors?

By a simple trick, can encode
continuous arguments too

$$f(x) \approx f(0.000000) =$$

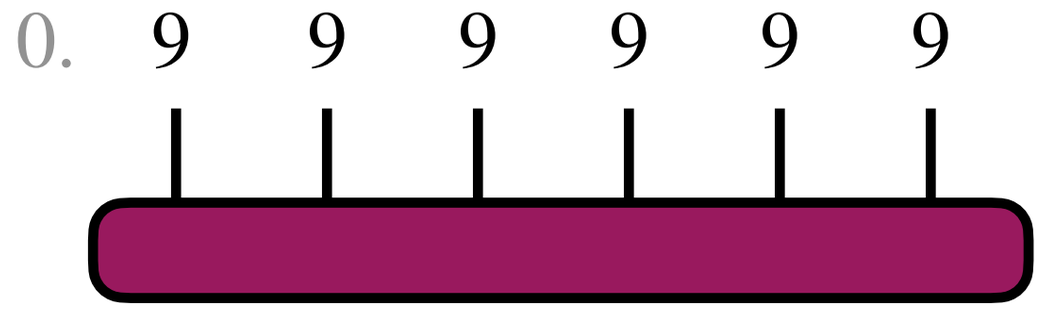


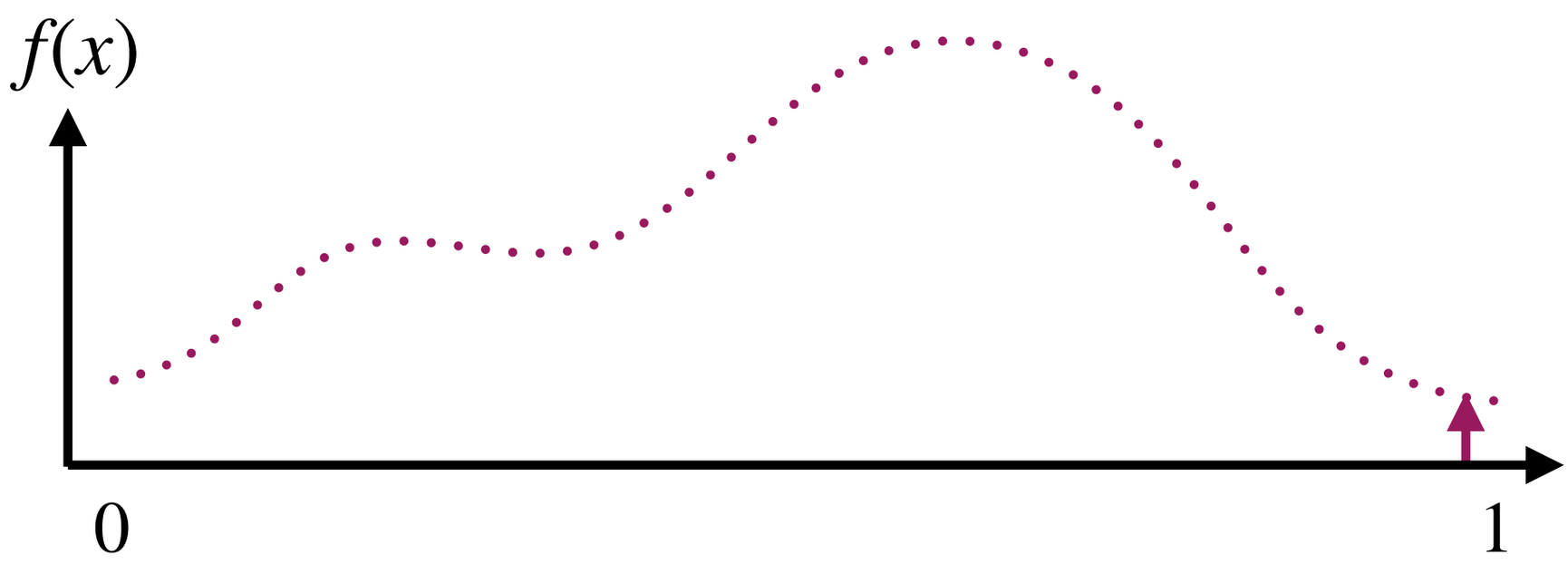


Why Tensors?

By a simple trick, can encode
continuous arguments too

$$f(x) \approx f(0.999999) =$$

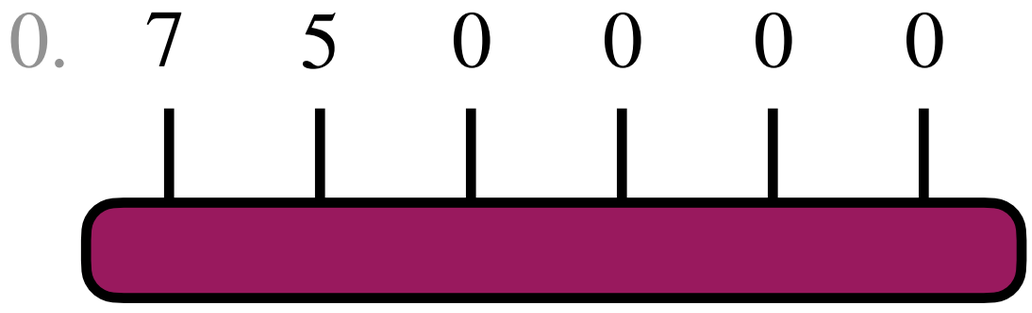


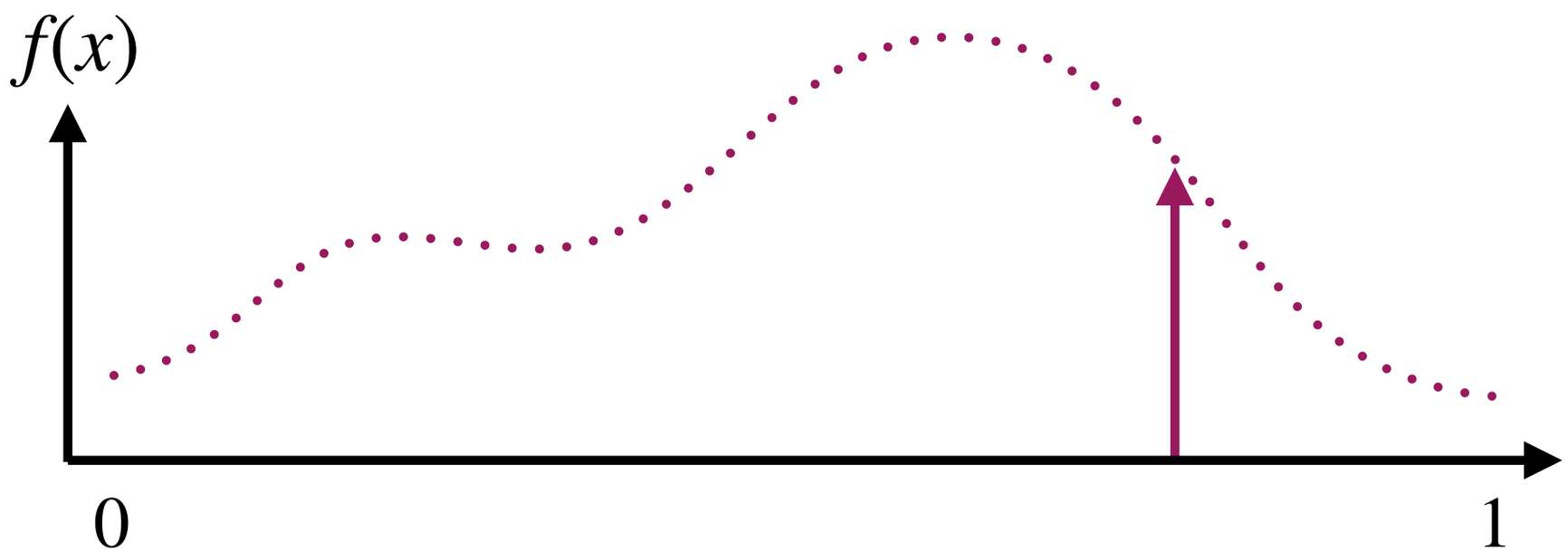


Why Tensors?

By a simple trick, can encode
continuous arguments too

$$f(x) \approx f(0.750000) =$$

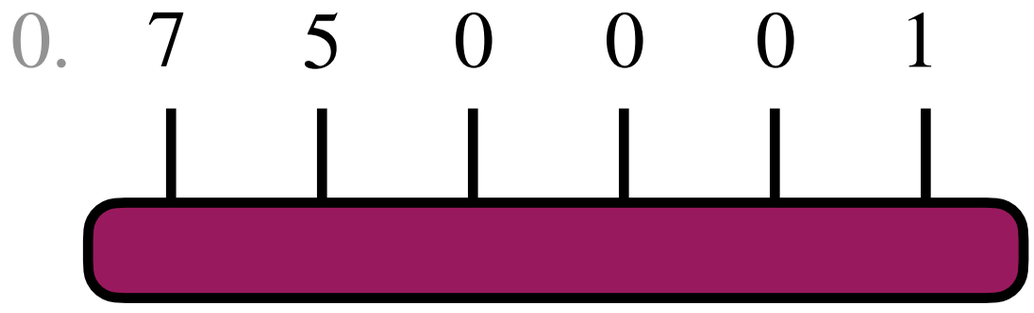


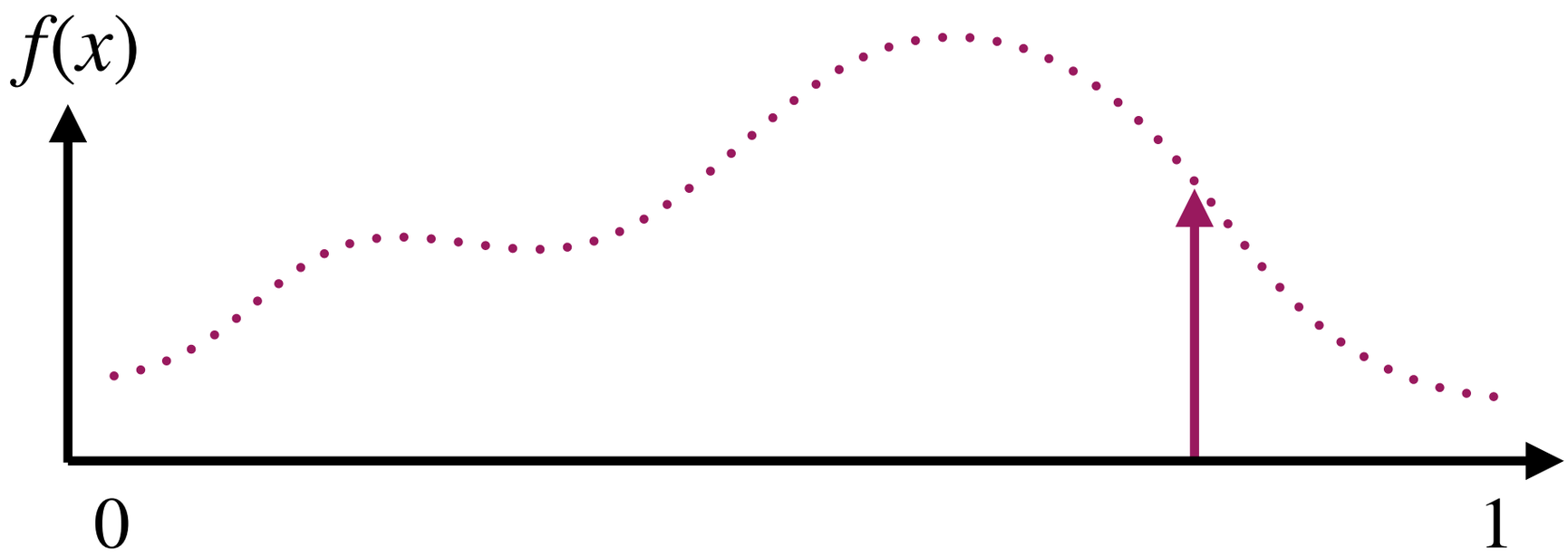


Why Tensors?

By a simple trick, can encode
continuous arguments too

$$f(x) \approx f(0.750001) =$$

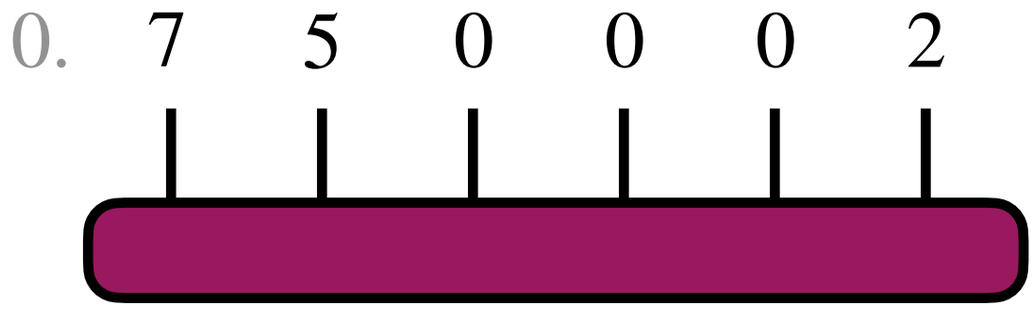


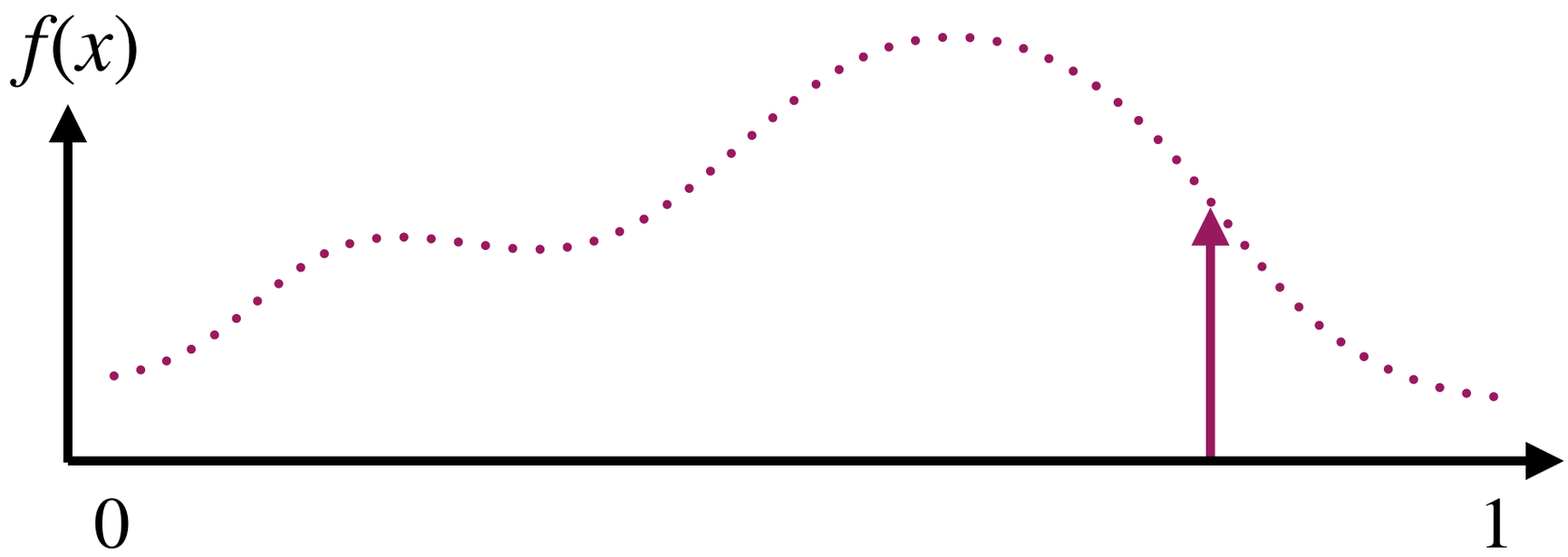


Why Tensors?

By a simple trick, can encode
continuous arguments too

$$f(x) \approx f(0.750002) =$$

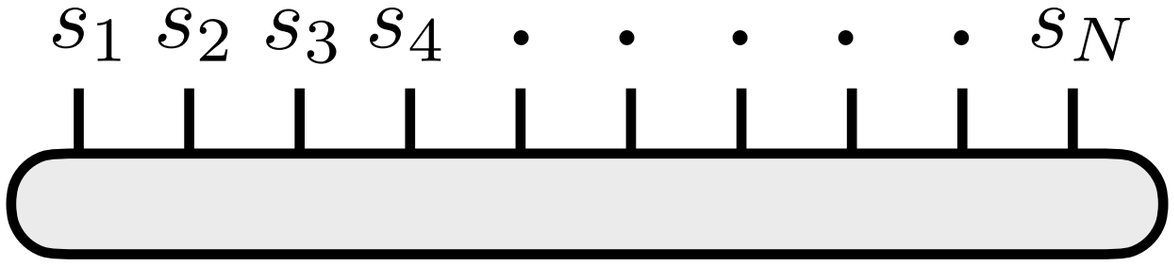




Classical Methods for Quantum Systems

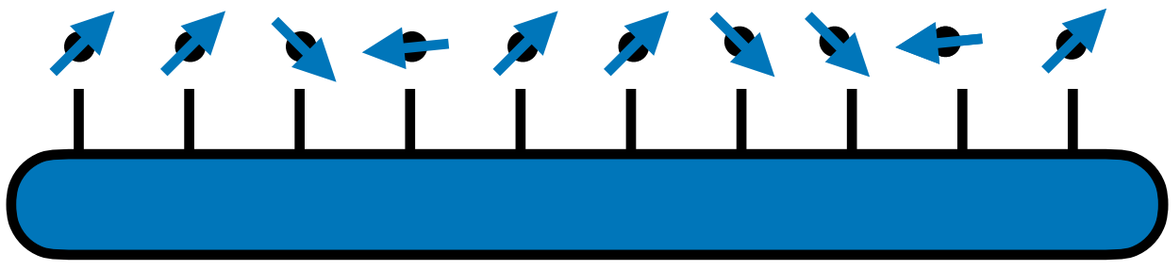
N-index tensor exponential to store

$$T^{s_1 s_2 s_3 \cdots s_N} =$$



Tensor version of "many-body problem"

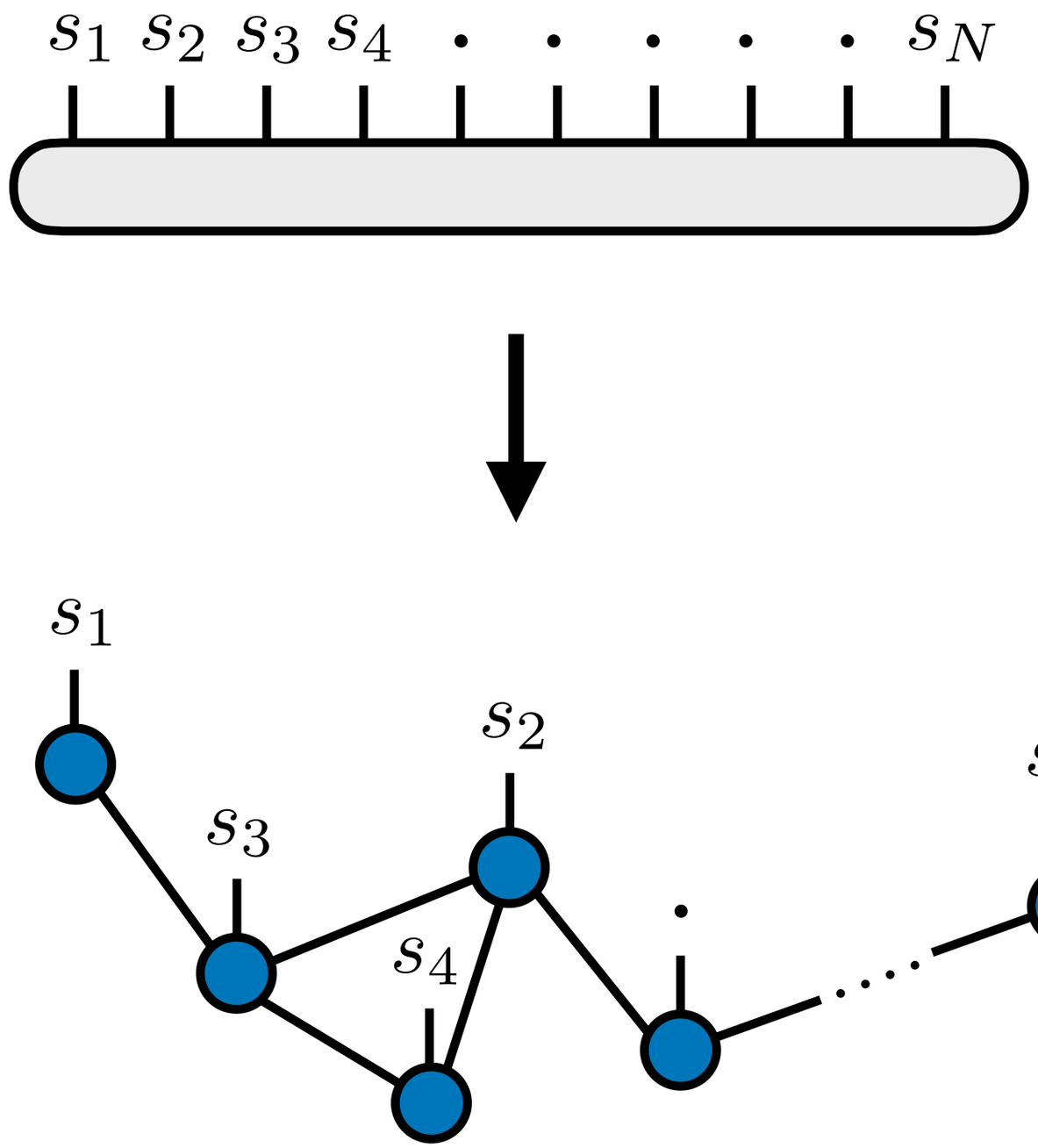
$$|\Psi\rangle =$$



Curse of Dimensionality

Tensor networks give a way to break the curse

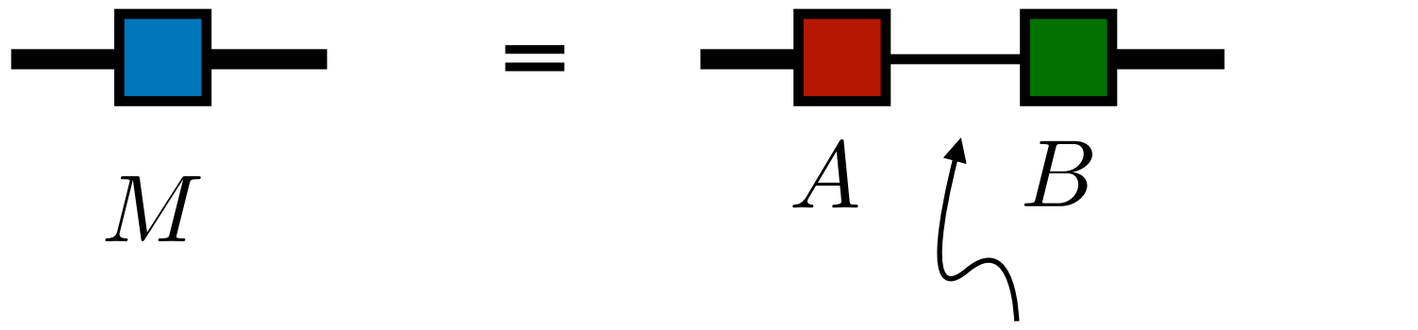
$$T^{s_1 s_2 s_3 \cdots s_N} =$$

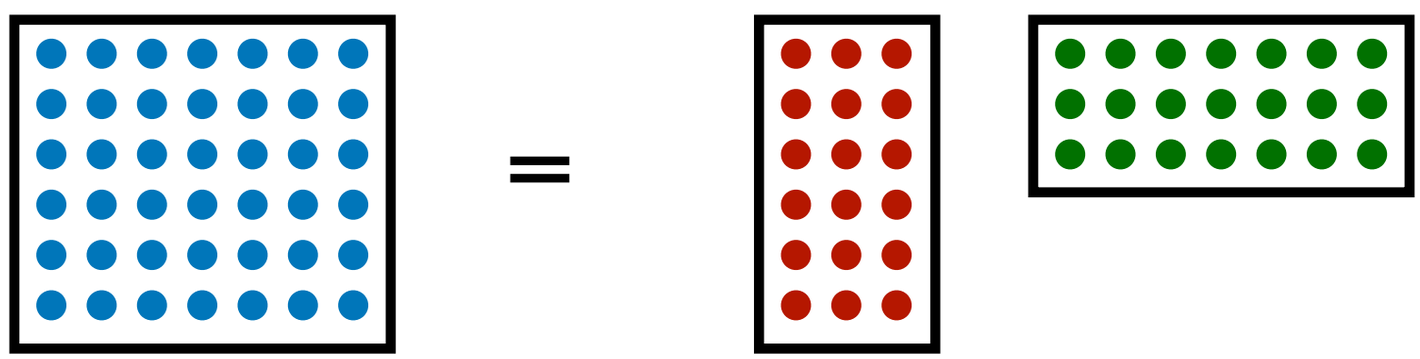


$$T^{s_1 s_2 s_3 \cdots s_N} \approx$$

Low-rank Structure

Finding low-rank structure solved for matrices

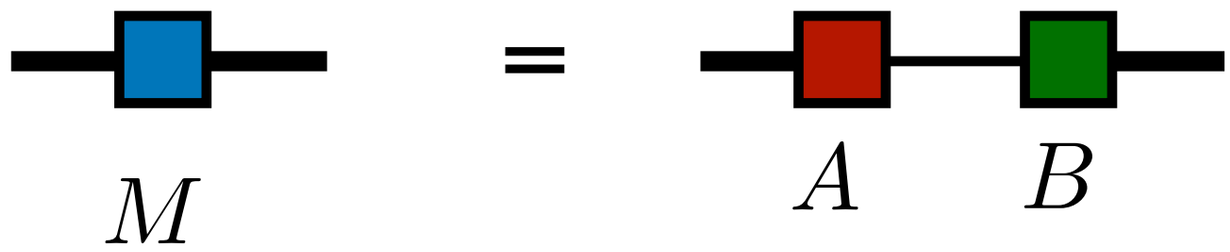


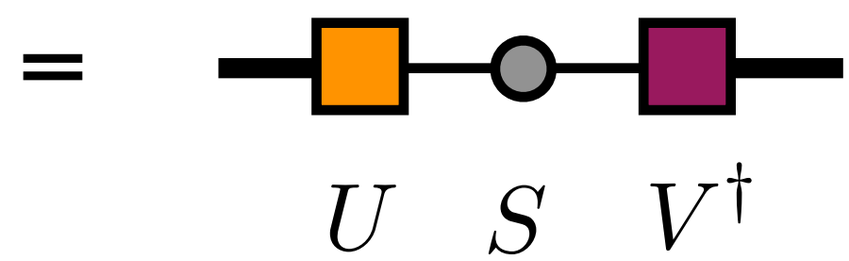


Solved by singular value decomposition (SVD)

Low-rank Structure

Finding low-rank structure solved for matrices





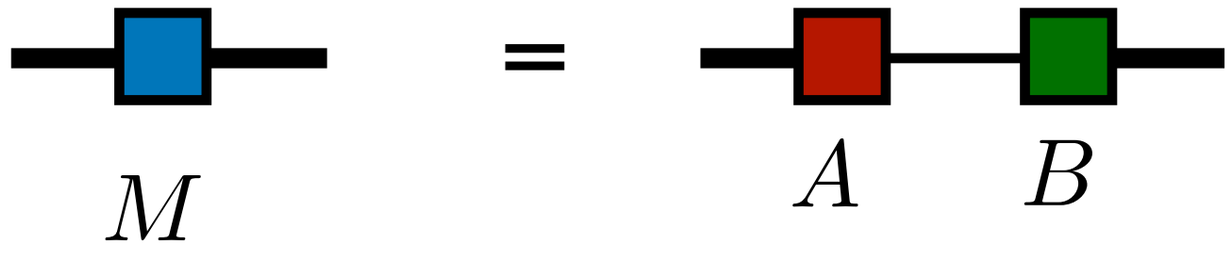
$$A = U\sqrt{S}$$

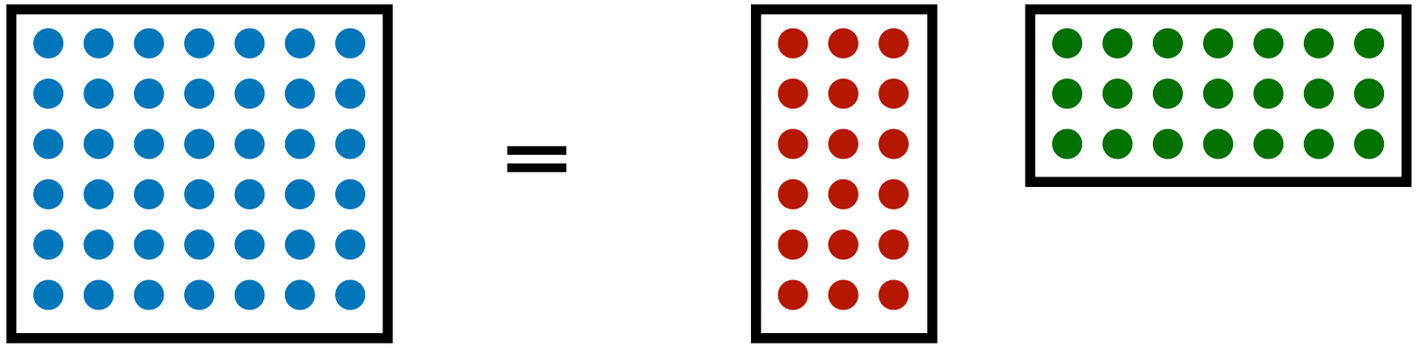
$$B = \sqrt{S}V^\dagger$$

Solved by singular value decomposition (SVD)

Low-rank Structure

Low-rank saves **memory** and **compute**



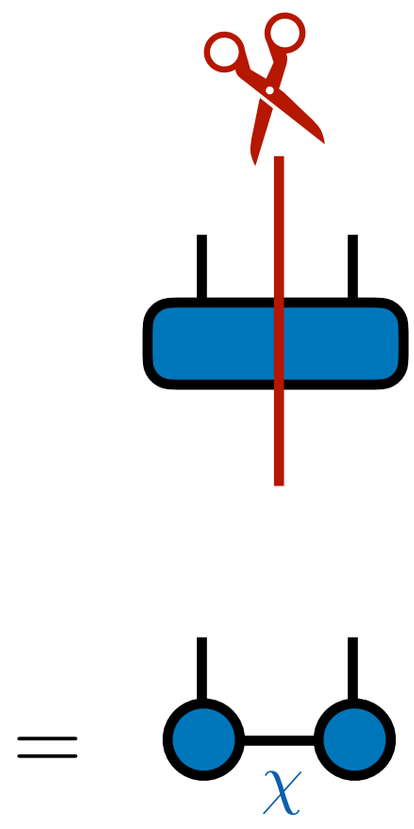


Many parameters

Few parameters

Classical Methods for Quantum Systems

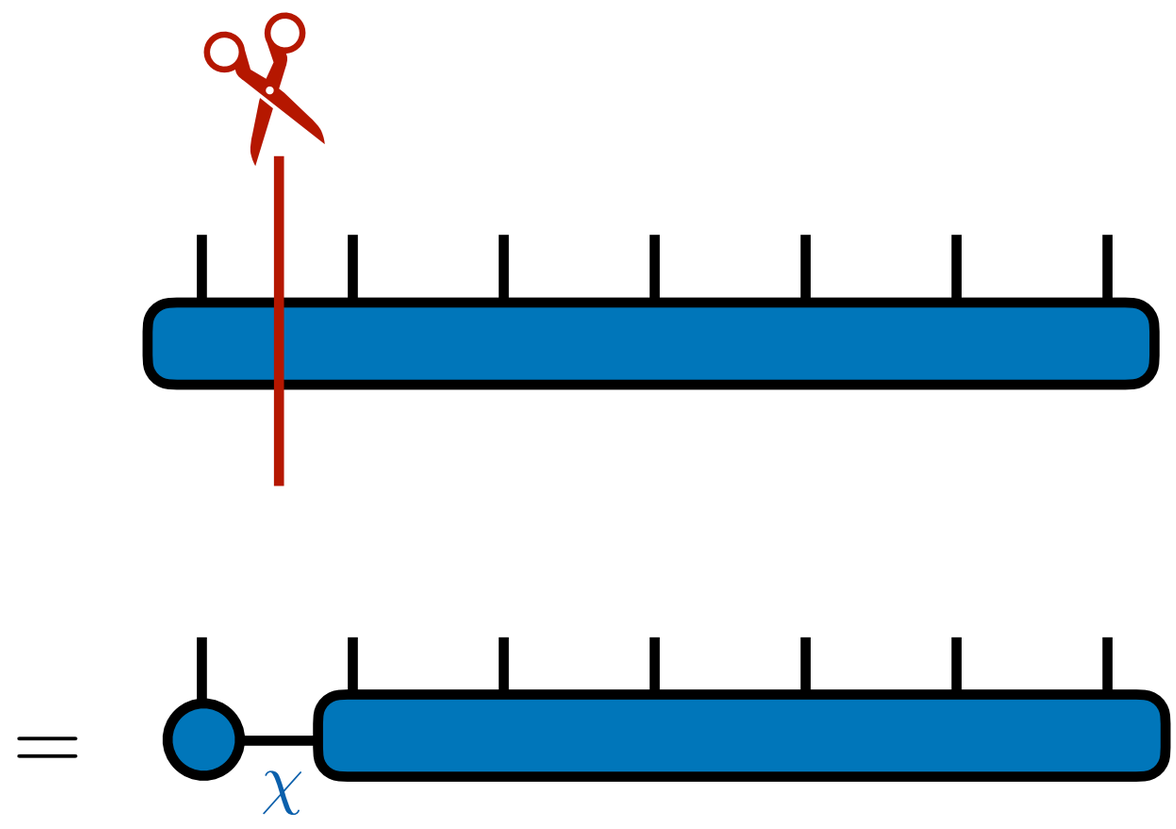
Just as factorizing a matrix reduces cost
(memory and compute)



χ is matrix rank

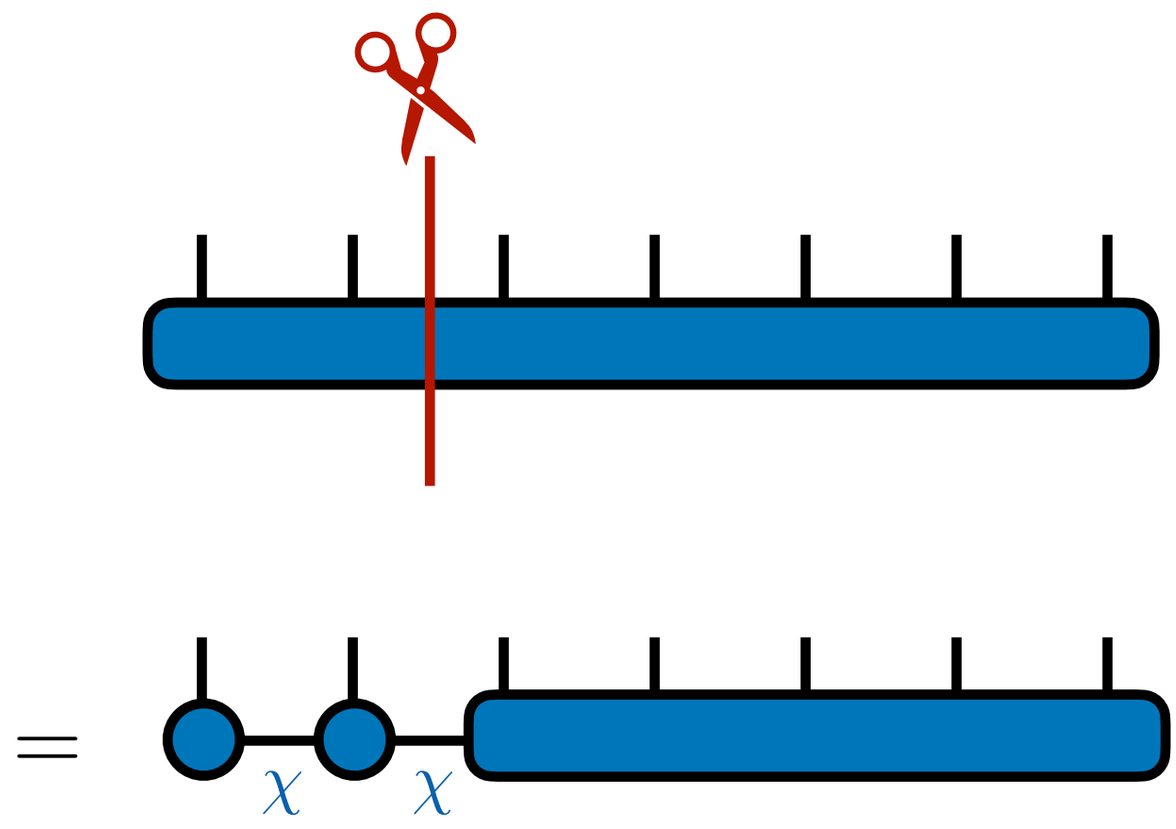
Classical Methods for Quantum Systems

Can recursively factor (compress) a tensor as well



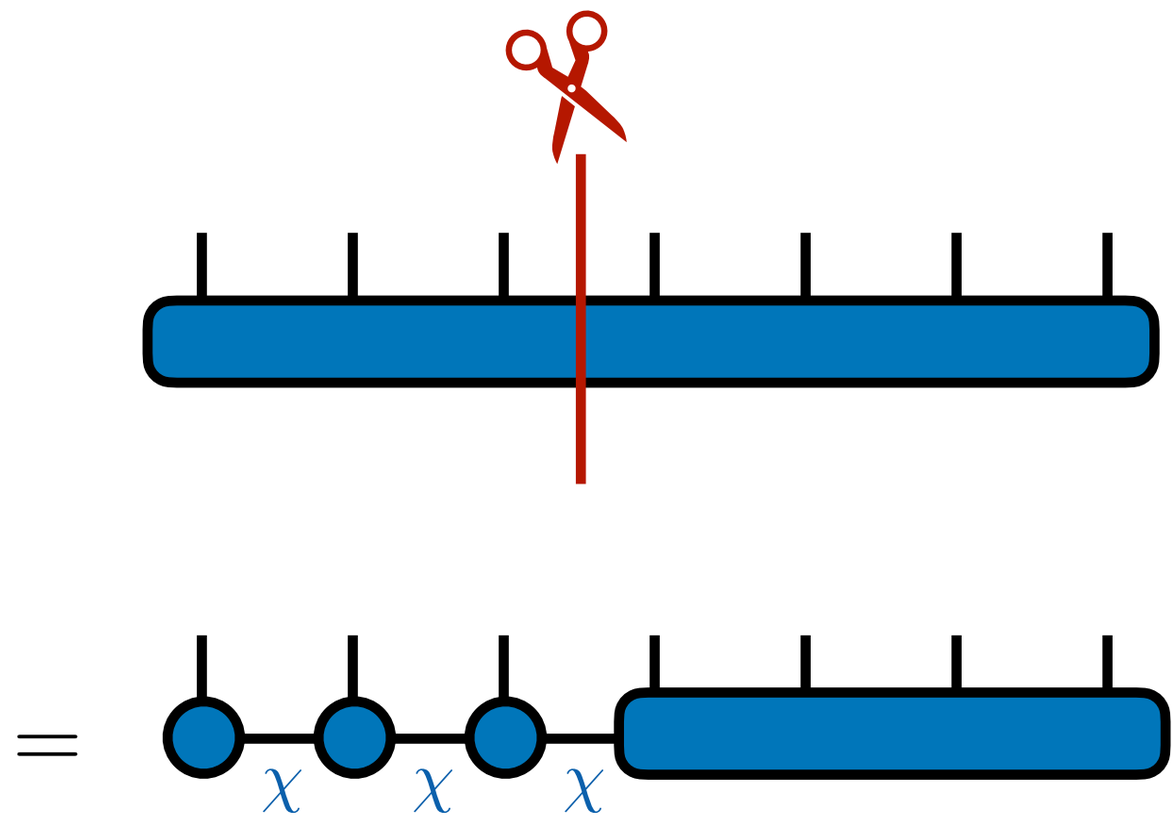
Classical Methods for Quantum Systems

Can recursively factor (compress) a tensor as well



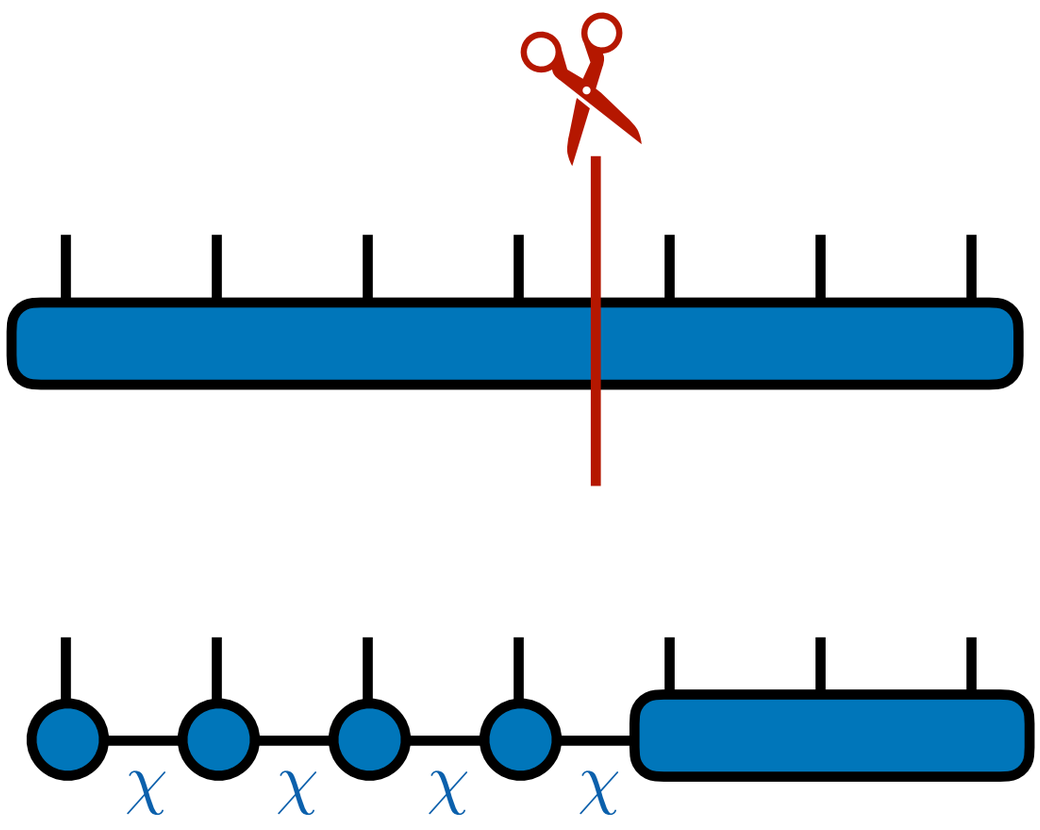
Classical Methods for Quantum Systems

Can recursively factor (compress) a tensor as well



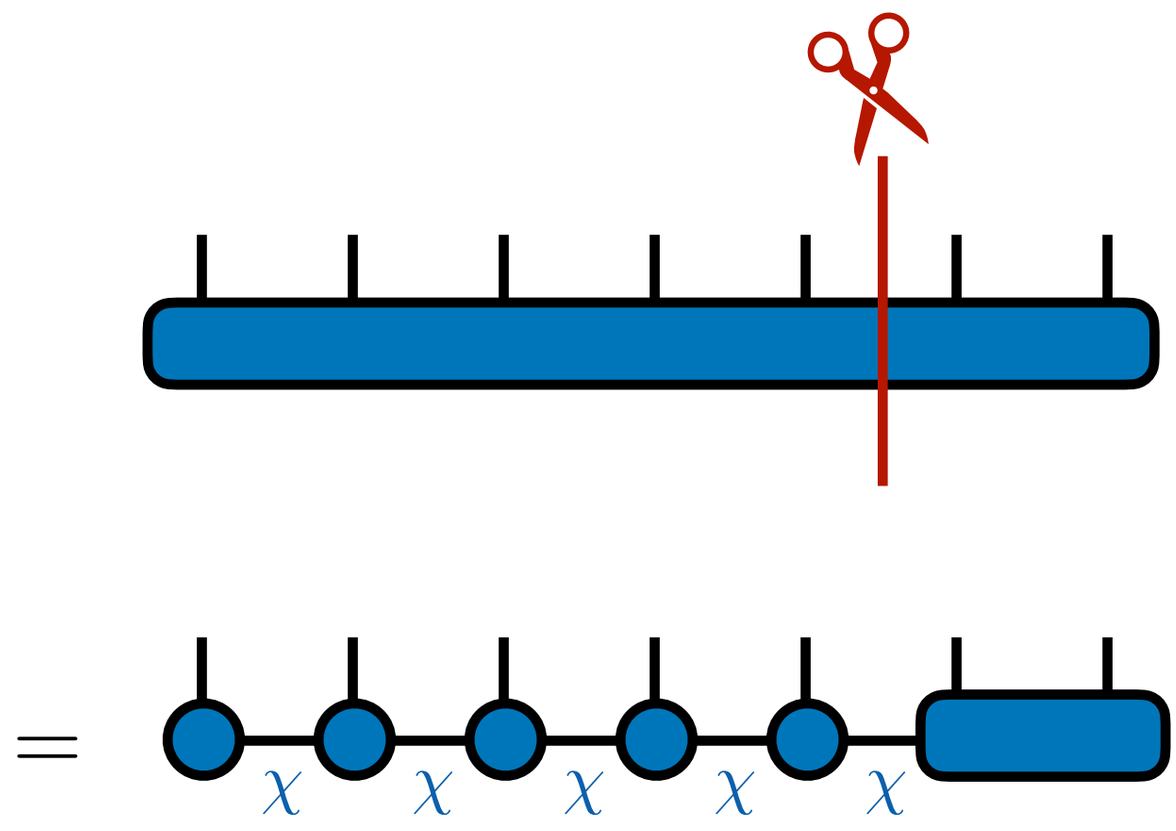
Classical Methods for Quantum Systems

Can recursively factor (compress) a tensor as well



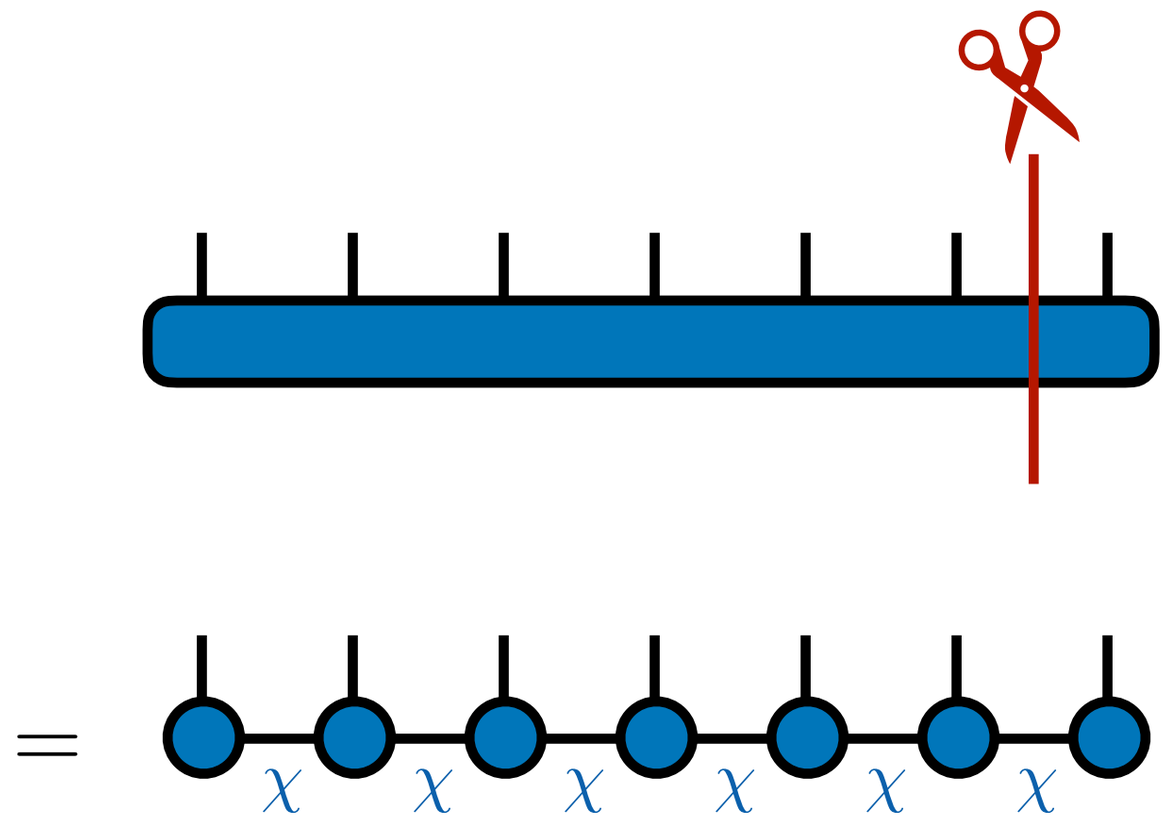
Classical Methods for Quantum Systems

Can recursively factor (compress) a tensor as well



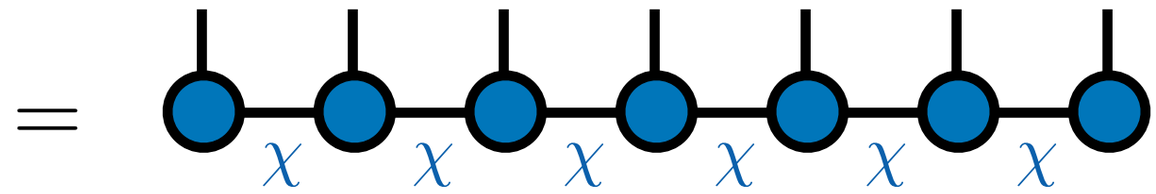
Classical Methods for Quantum Systems

Can recursively factor (compress) a tensor as well



Classical Methods for Quantum Systems

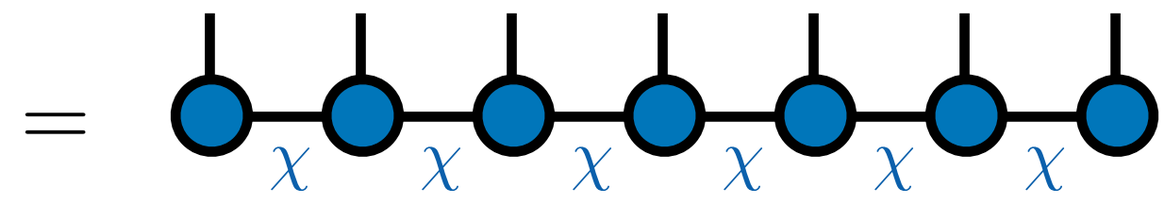
Result is a "matrix product state" (MPS)



If "ranks" χ small, save on memory and compute

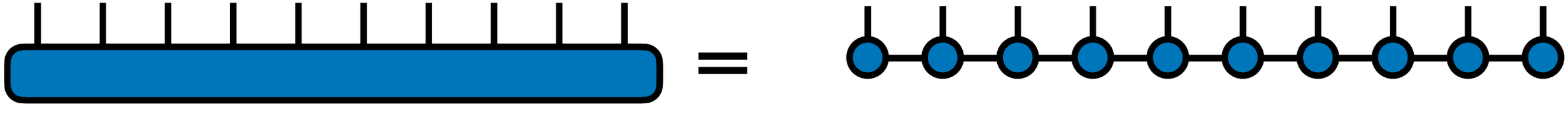
Classical Methods for Quantum Systems

Result is a "matrix product state" (MPS)



Technically, scaling is χ^2 memory , χ^3 compute

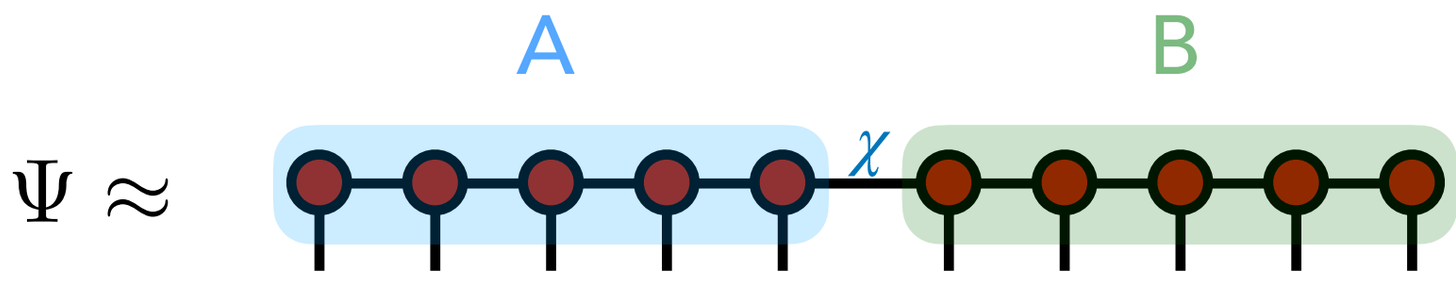
Matrix product state (MPS) tensor network



Can view as multi-SVD of a tensor

Or special class or subspace of tensors
(low-rank subspace)

Entanglement of MPS tensor network

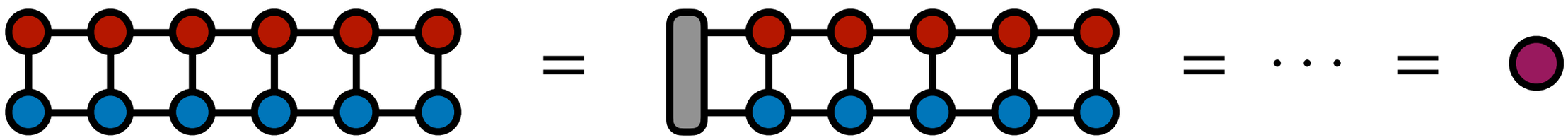


Partition MPS into left (A) and right (B) exactly a rank χ matrix factorization

Entanglement bounded by $\log(\chi)$

MPS efficiently capture **moderately-entangled** states

Inner product of two MPS tensors



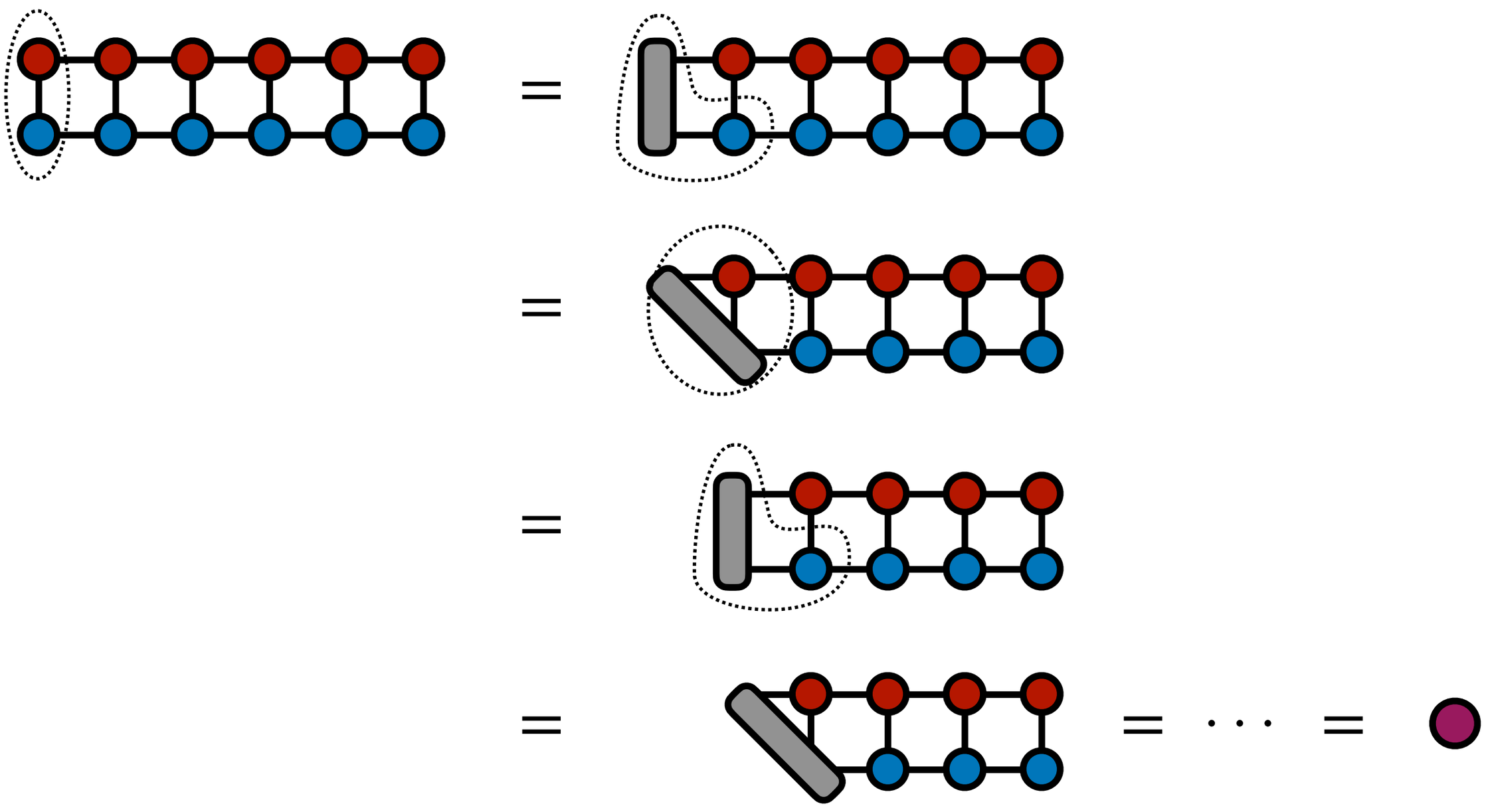
If each MPS represents tensor with 40 indices of dimension 2

Then above algorithm computes dot product of two 'vectors' of a *trillion* entries each

Takes ~ 10ms for bond dimension $\chi = 100$

More detailed tensor network algorithm

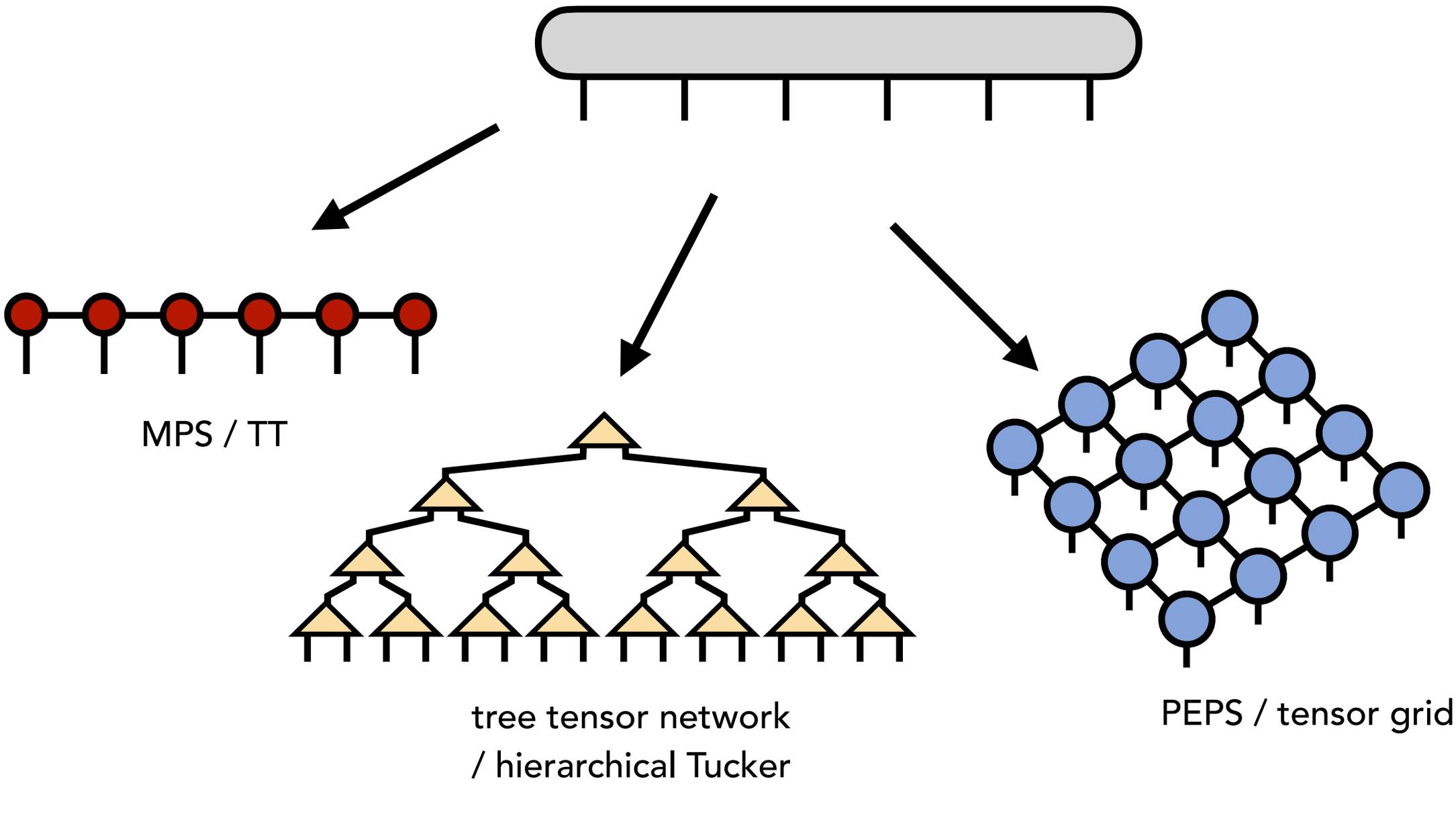
Inner product of two MPS tensors



Cost $\sim \chi^3$, memory usage $\sim \chi^2$

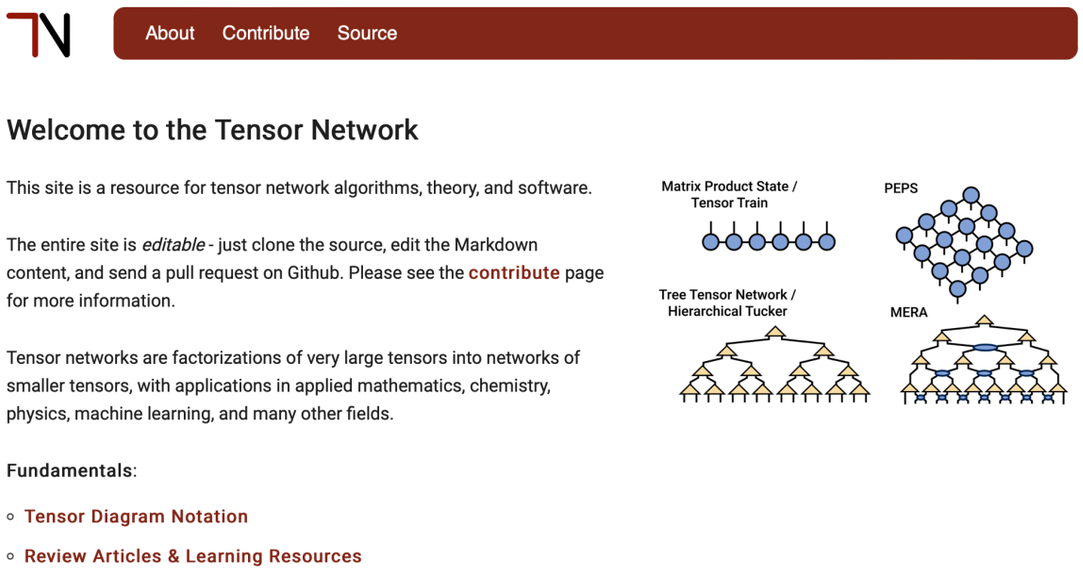
Tensor Networks

There are other tensor networks too,
with their own algorithms and degrees of expressive power

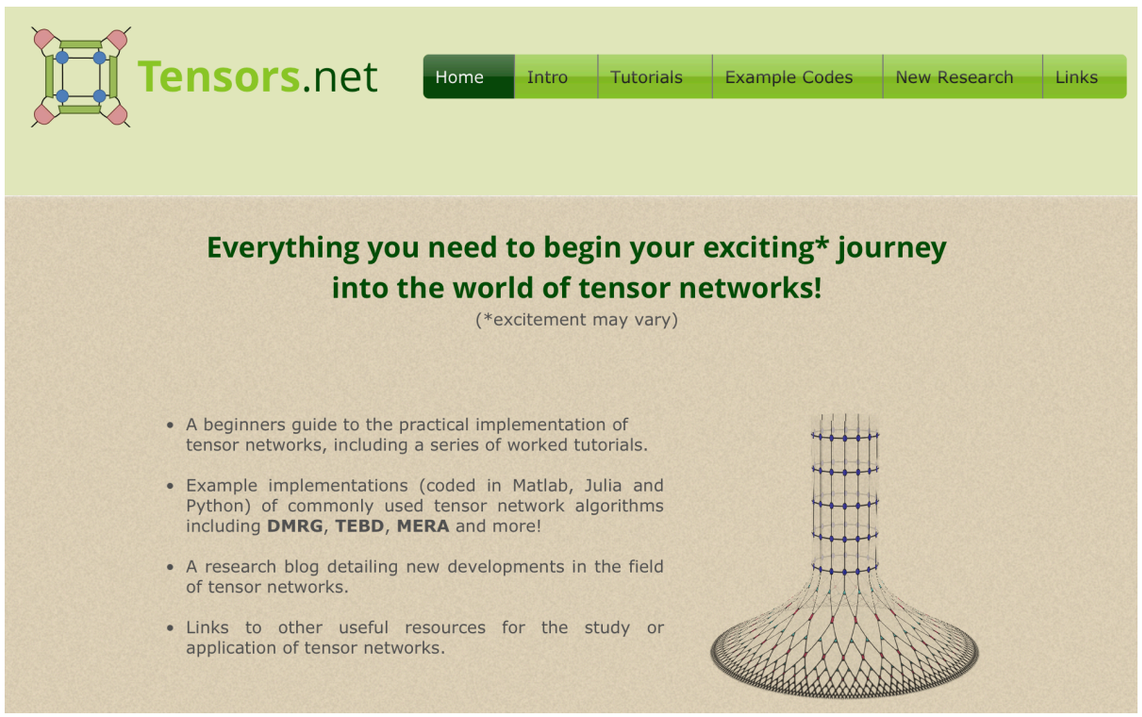


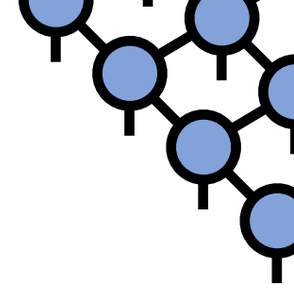
More Resources about Tensor Networks

- TensorNetwork.org: list of review articles, video lectures, pages with details of algorithms



- Tensors.net: tutorials and sample codes





Plan for Today

- Tensors and Tensor Networks

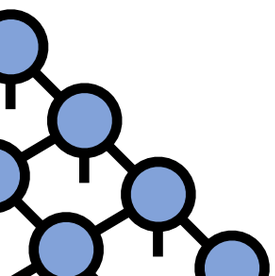
- Quantum Circuits as Tensor Networks

- Intro to the ITensor Software

- Algorithms for Tensor Networks (MPS and MPO)

- Using ITensor on GPU

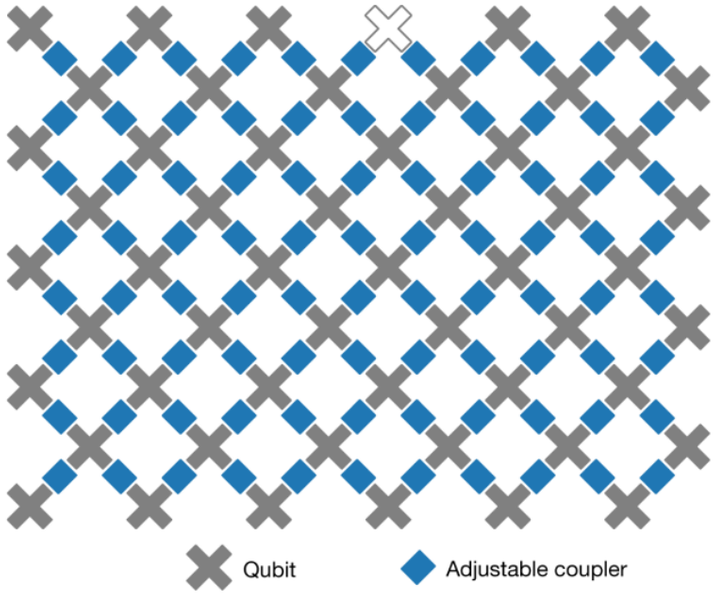
- Future of Tensor Networks: Functions & Machine Learning

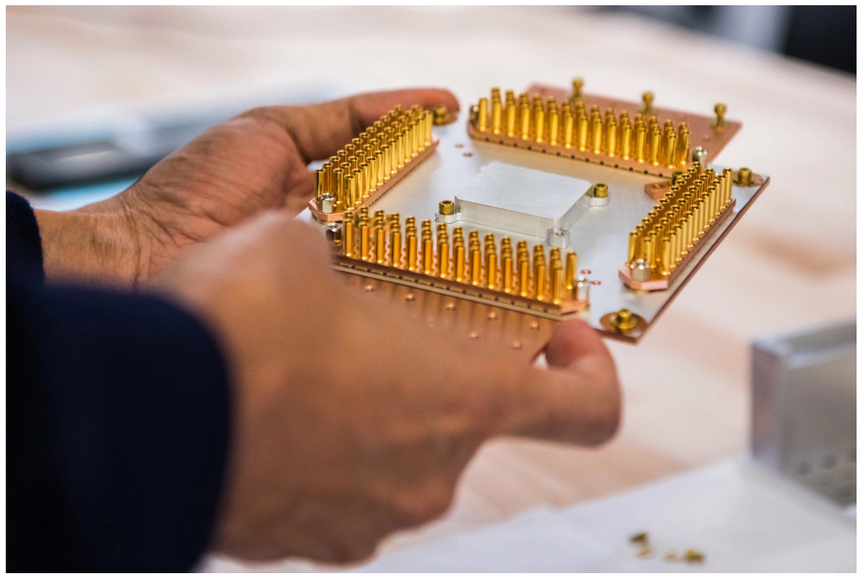


Quantum Computing as a Tensor Network

A quantum computer is a machine of *quantum bits*

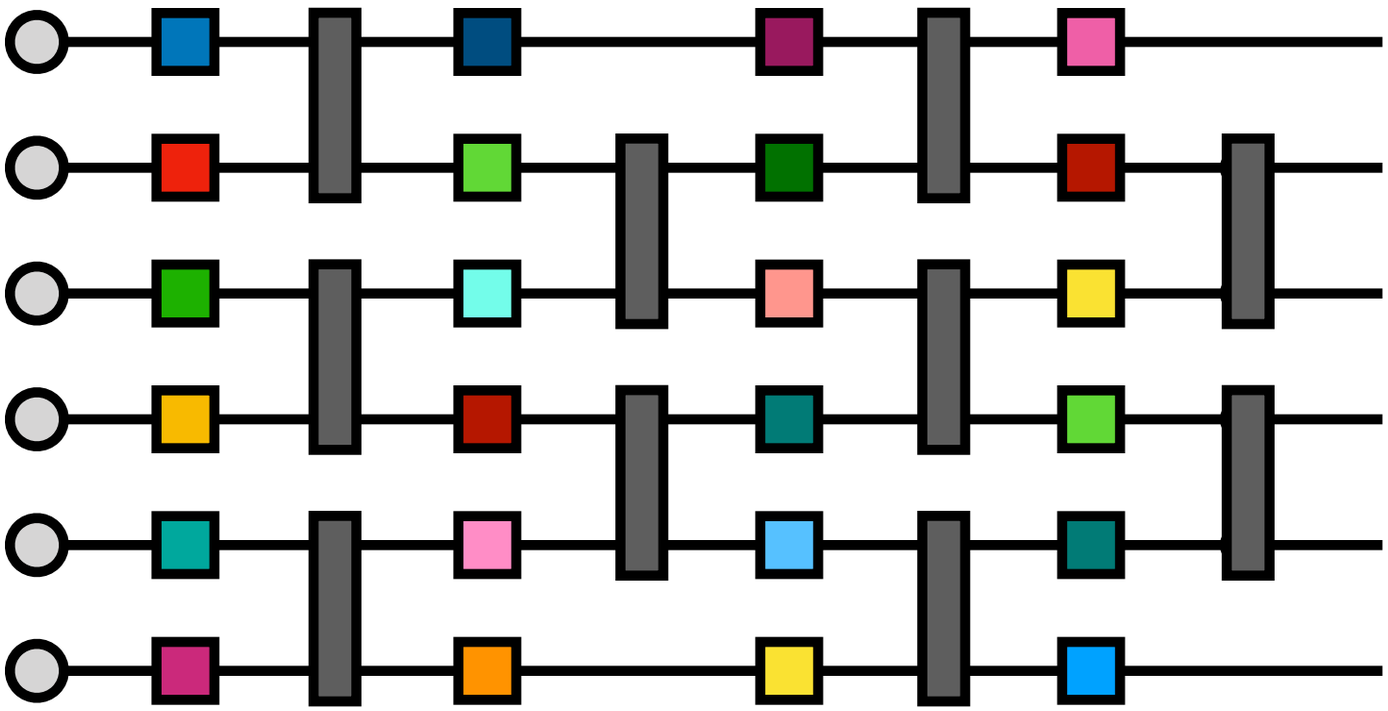
Carries out instructions, exploring an exponentially large state space





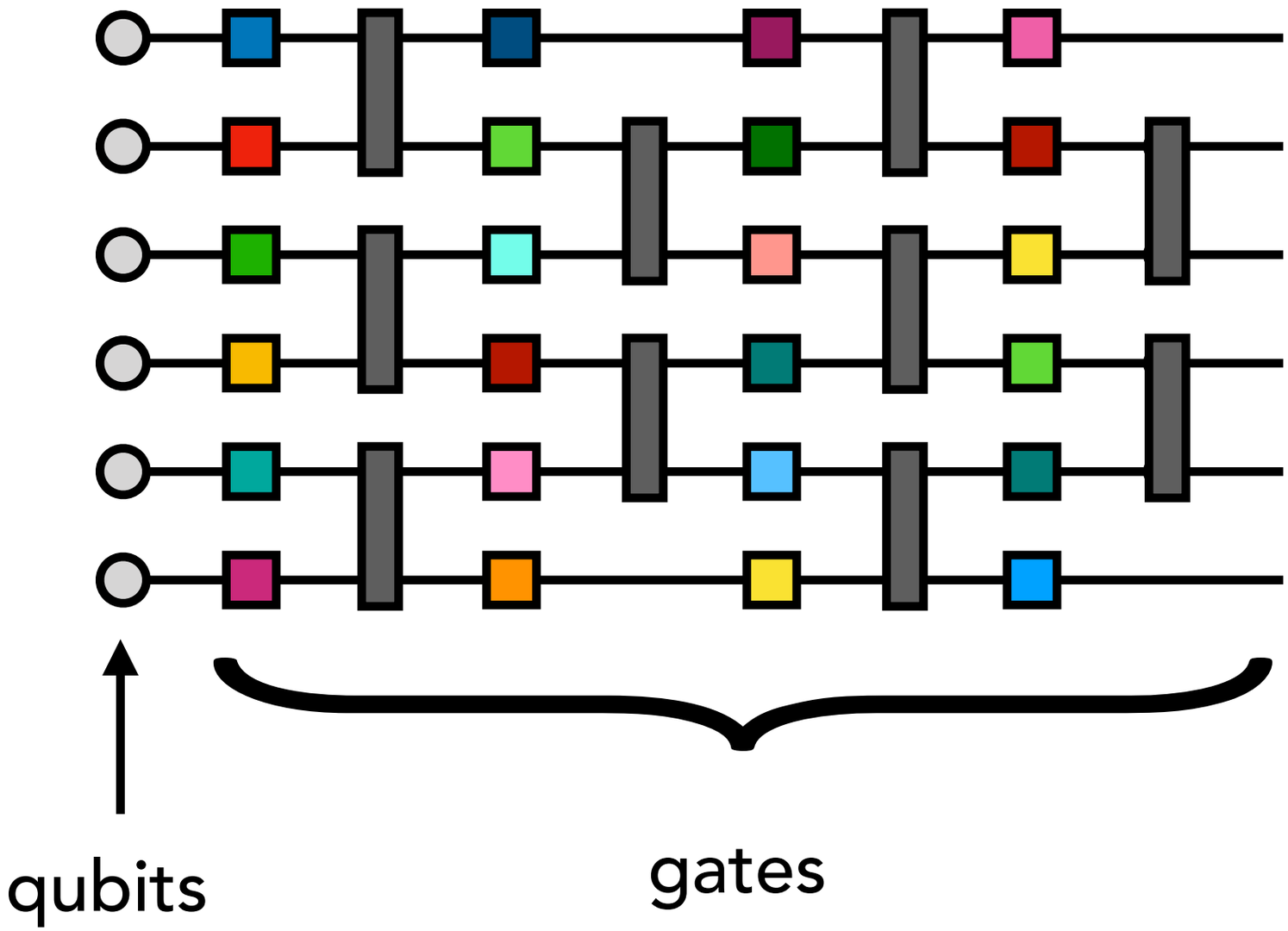
Quantum Circuits

"Program" for a quantum computer is a *quantum circuit*



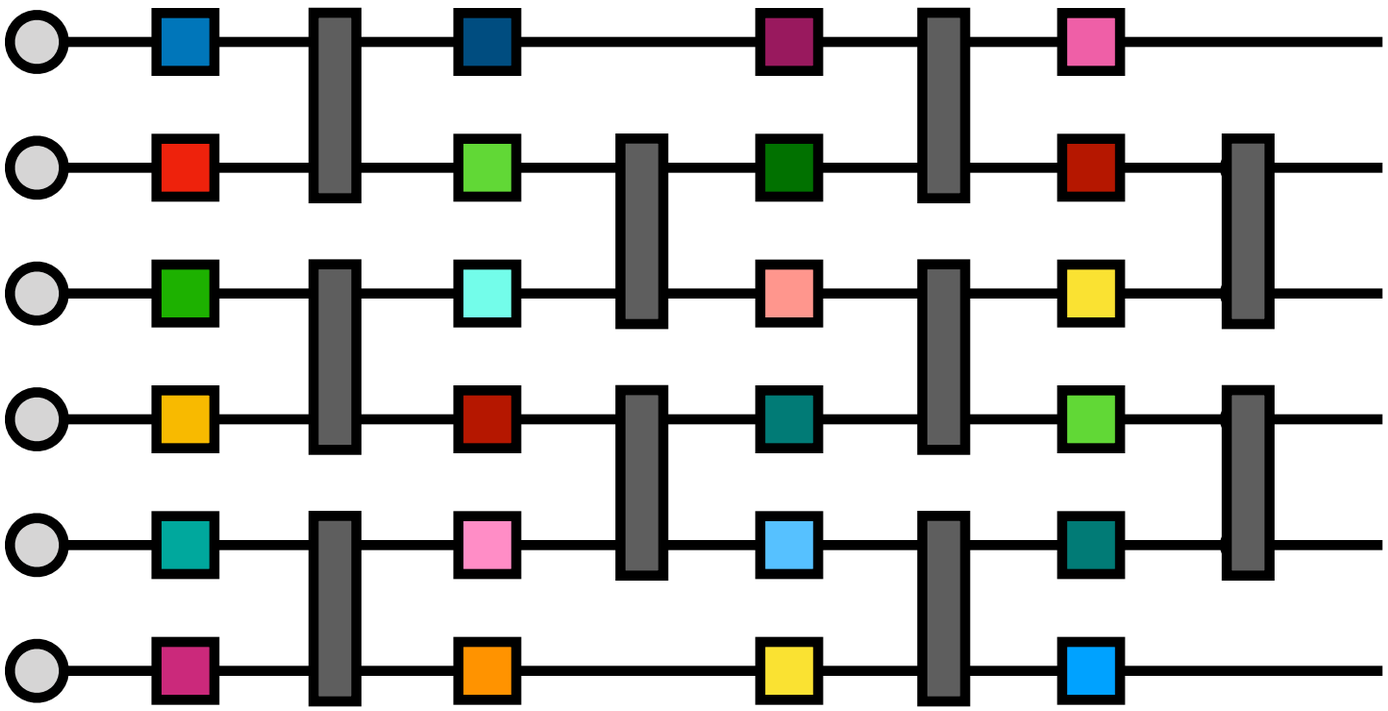
Quantum Circuits

"Program" for a quantum computer is a *quantum circuit*



Quantum Circuits

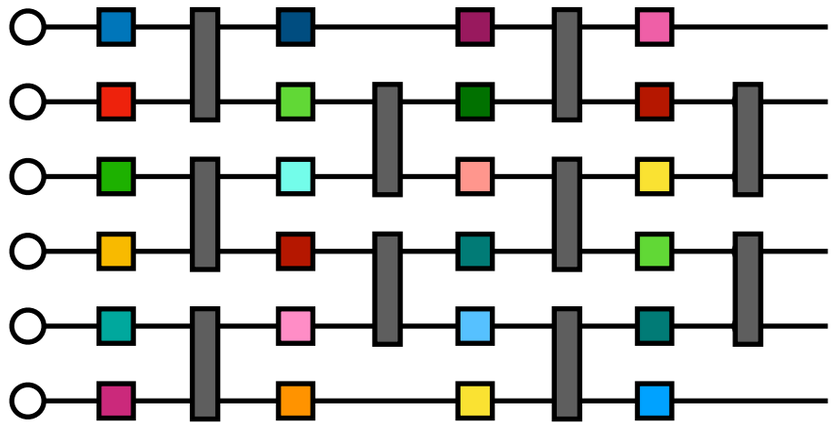
Every quantum circuit in one-to-one correspondence with a tensor network (same diagram)

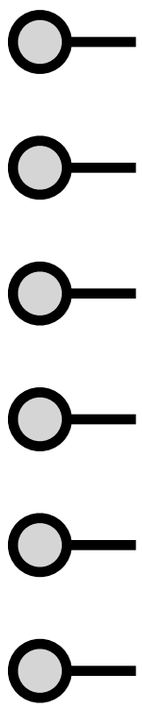


Let's see how...

Quantum Circuits

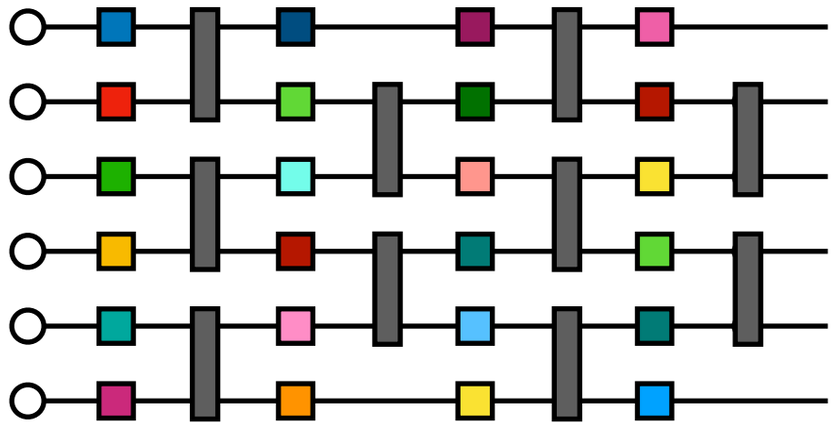
Vectors representing initial state





$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

Quantum Circuits



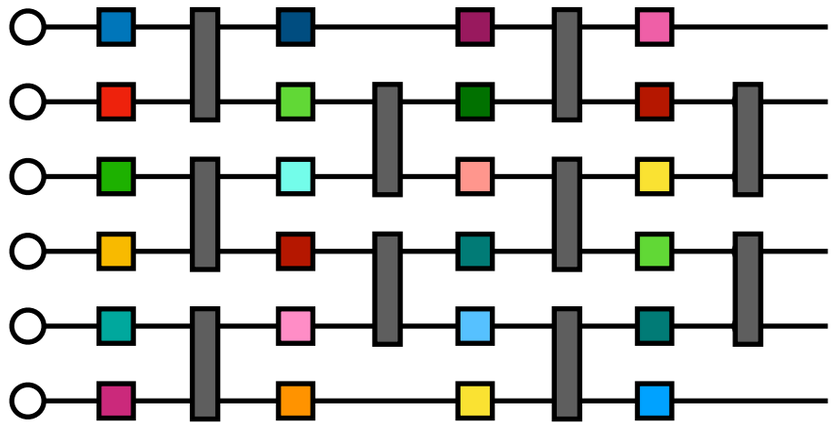
Matrices representing single-qubit gates

$$s \quad \quad s' \quad = U_{ss'}$$

For example, Hadamard gate

$$U = H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Quantum Circuits

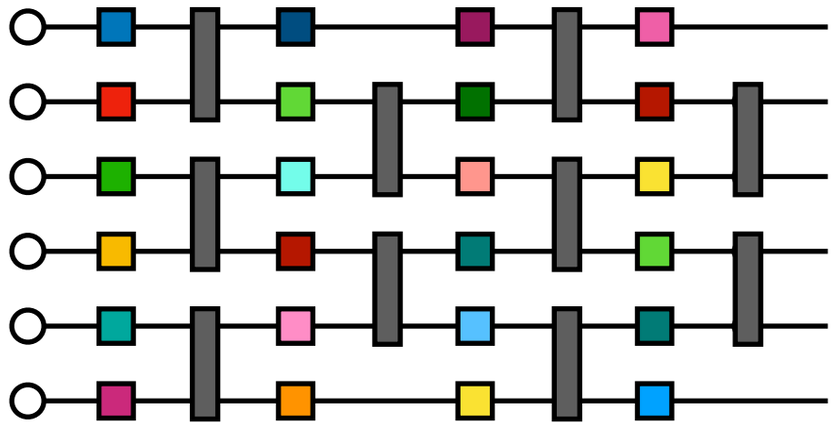


Matrices representing single-qubit gates

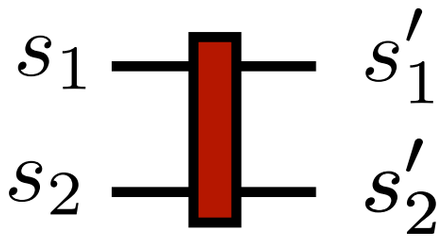
For example, NOT gate

$$U = NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Quantum Circuits



Tensors representing two-qubit gates



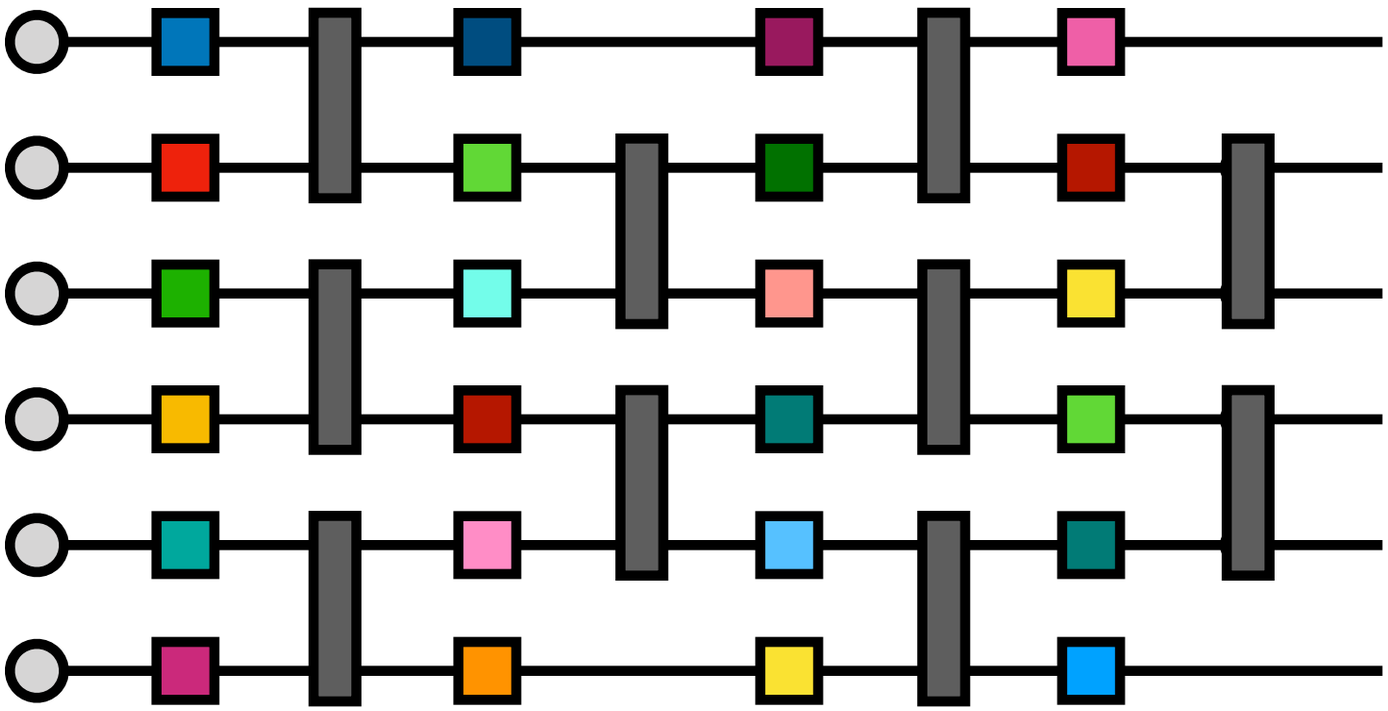
$$= U_{s_1 s_2}^{s_1' s_2'}$$

For example, controlled not (CNOT) gate

$$U = CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

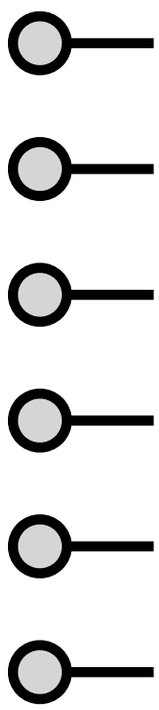
Quantum Circuits

The state after executing the quantum circuit is given by contracting all the tensors



Quantum Circuits

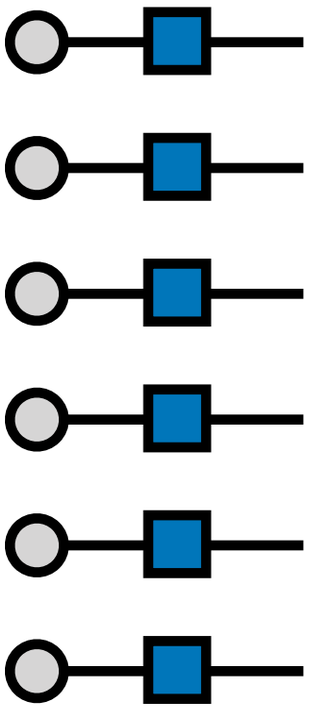
Example:



$$|\Psi\rangle = \begin{bmatrix}1\\0\end{bmatrix}_1 \begin{bmatrix}1\\0\end{bmatrix}_2 \begin{bmatrix}1\\0\end{bmatrix}_3 \begin{bmatrix}1\\0\end{bmatrix}_4 \begin{bmatrix}1\\0\end{bmatrix}_5 \begin{bmatrix}1\\0\end{bmatrix}_6$$

Quantum Circuits

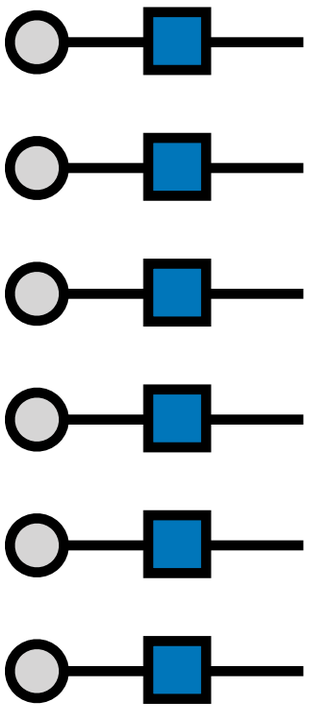
Act with Hadamard gates



$$|\Psi\rangle = \begin{bmatrix}1\\1\end{bmatrix}_1 \begin{bmatrix}1\\1\end{bmatrix}_2 \begin{bmatrix}1\\1\end{bmatrix}_3 \begin{bmatrix}1\\1\end{bmatrix}_4 \begin{bmatrix}1\\1\end{bmatrix}_5 \begin{bmatrix}1\\1\end{bmatrix}_6 \Big/ \sqrt{2}^6$$

Quantum Circuits

Act with Hadamard gates

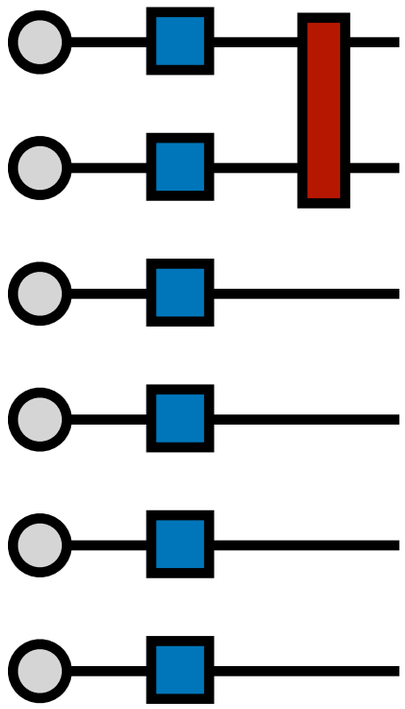


Unentangled state,
massive superposition

$$|\Psi\rangle = \begin{bmatrix}1\\1\end{bmatrix}_1 \begin{bmatrix}1\\1\end{bmatrix}_2 \begin{bmatrix}1\\1\end{bmatrix}_3 \begin{bmatrix}1\\1\end{bmatrix}_4 \begin{bmatrix}1\\1\end{bmatrix}_5 \begin{bmatrix}1\\1\end{bmatrix}_6 \Big/ \sqrt{2}^6$$

Quantum Circuits

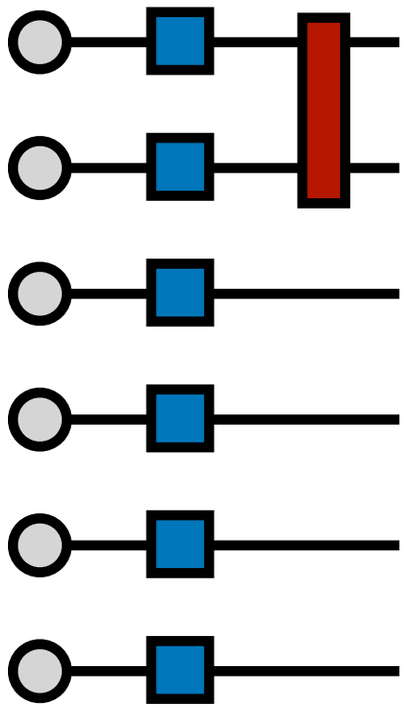
Act with CZ gate on qubits 1 & 2



$$|\Psi\rangle = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{1,2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{3} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{4} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{5} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{6} / \sqrt{2}^{6}$$

Quantum Circuits

Act with CZ gate on qubits 1 & 2

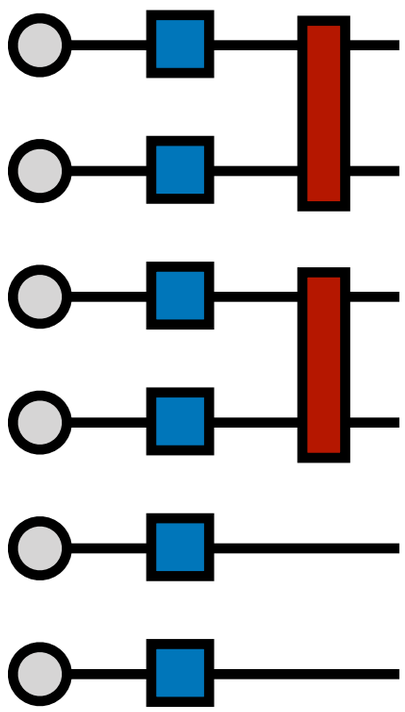


Entanglement
between qubits 1 & 2

$$|\Psi\rangle = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{1,2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{3} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{4} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{5} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{6} / \sqrt{2}^{6}$$

Quantum Circuits

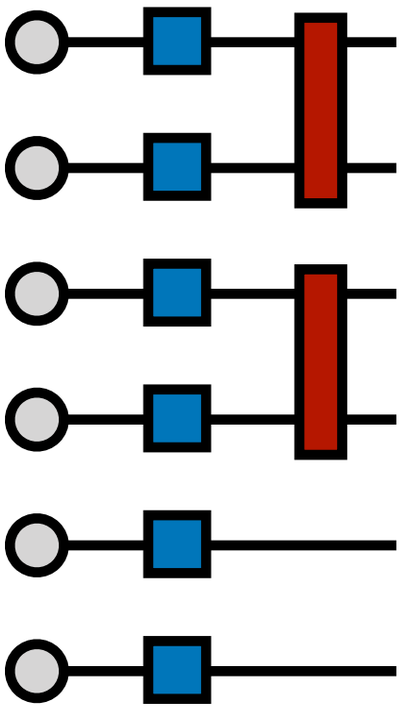
Act with CZ gate on qubits 3 & 4



$$|\Psi\rangle = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{1,2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{3,4} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{5} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{6} / \sqrt{2}^{6}$$

Quantum Circuits

Act with CZ gate on qubits 3 & 4

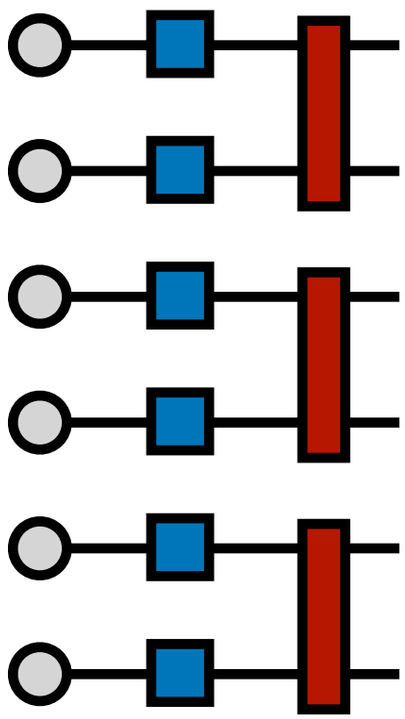


Entanglement
between qubits 3 & 4

$$|\Psi\rangle = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{1,2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{3,4} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{5} \begin{bmatrix} 1 \\ 1 \end{bmatrix}_{6} / \sqrt{2}^{6}$$

Quantum Circuits

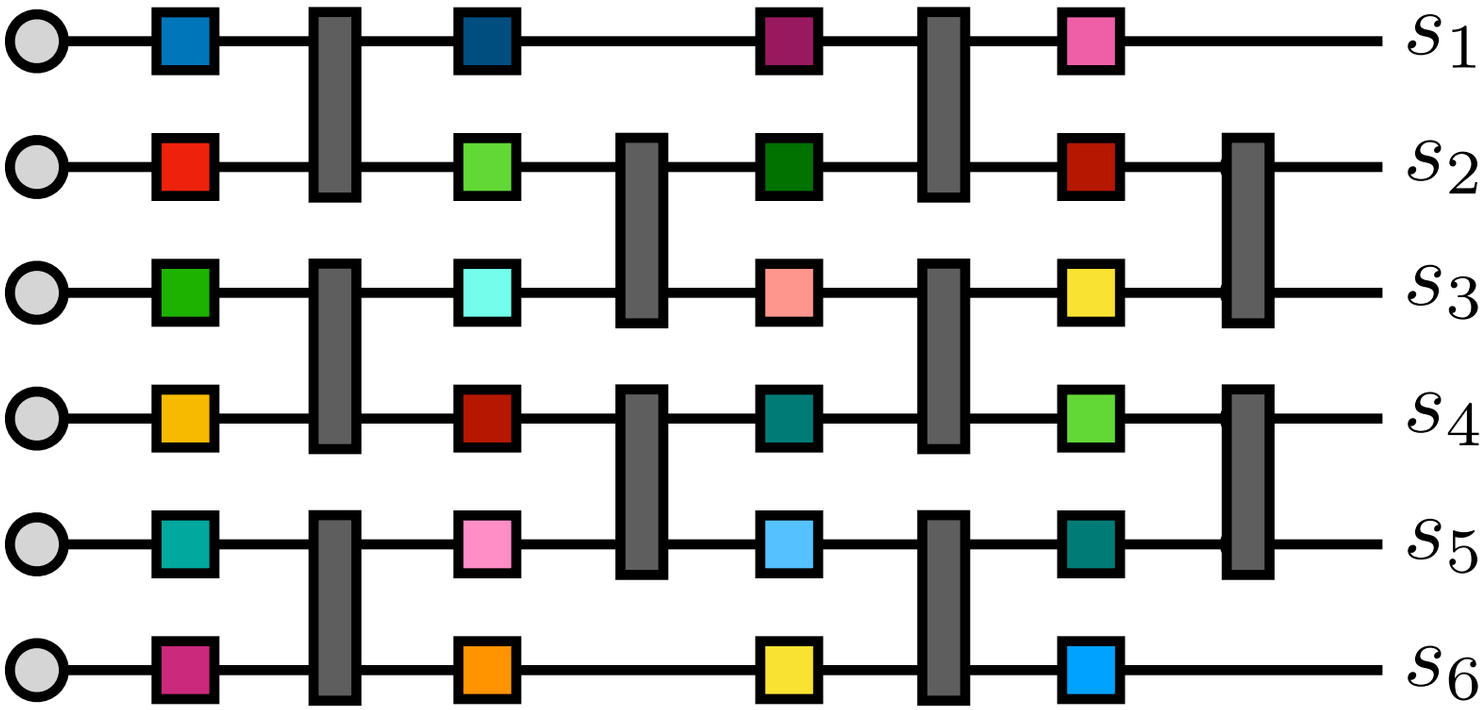
Act with CZ gate on qubits 5 & 6

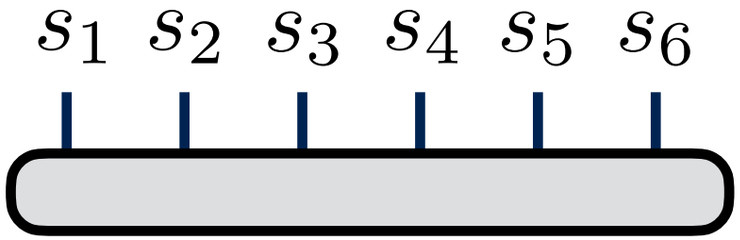


$$|\Psi\rangle = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{1,2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{3,4} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{5,6} / \sqrt{2}^{6}$$

Quantum Circuits

After enough gates, state can get complicated

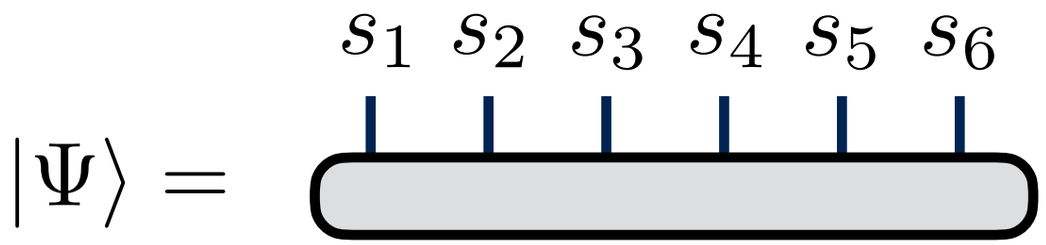


$|\Psi\rangle =$ 

complicated tensor defined by circuit (quantum state)

Quantum Circuits

Quantum computing forces you to encounter large tensors



complicated tensor
defined by circuit
(quantum state)

What if we could work with large tensors on a classical computer?

Tensor Network Algorithms

Tensor Network Algorithms

Most important aspect of tensor networks are
the algorithms

Two examples

- algorithm for finding ground states ("DMRG")[1,2]
- algorithm for applying gates ("TEBD")[3]

[1] White, PRL 69, 2863 (1992)
[2] Schollwöck, Annals of Phys. 326, 96 (2011)
[3] Vidal, PRL 91, 147902 (2003)

Tensor Network Algorithms

Seminal tensor network algorithm is DMRG
(density matrix renormalization group)

Given a Hamiltonian H

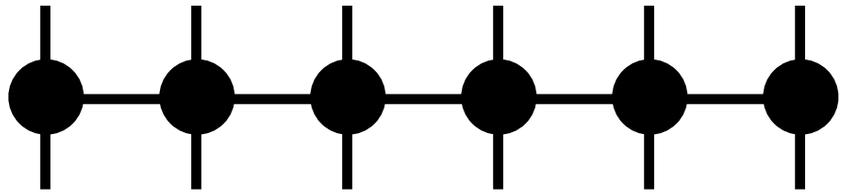
Finds **ground state** and its **energy**

$$\min_{\psi} \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} = E_0$$

Tensor Network Algorithms

DMRG algorithm

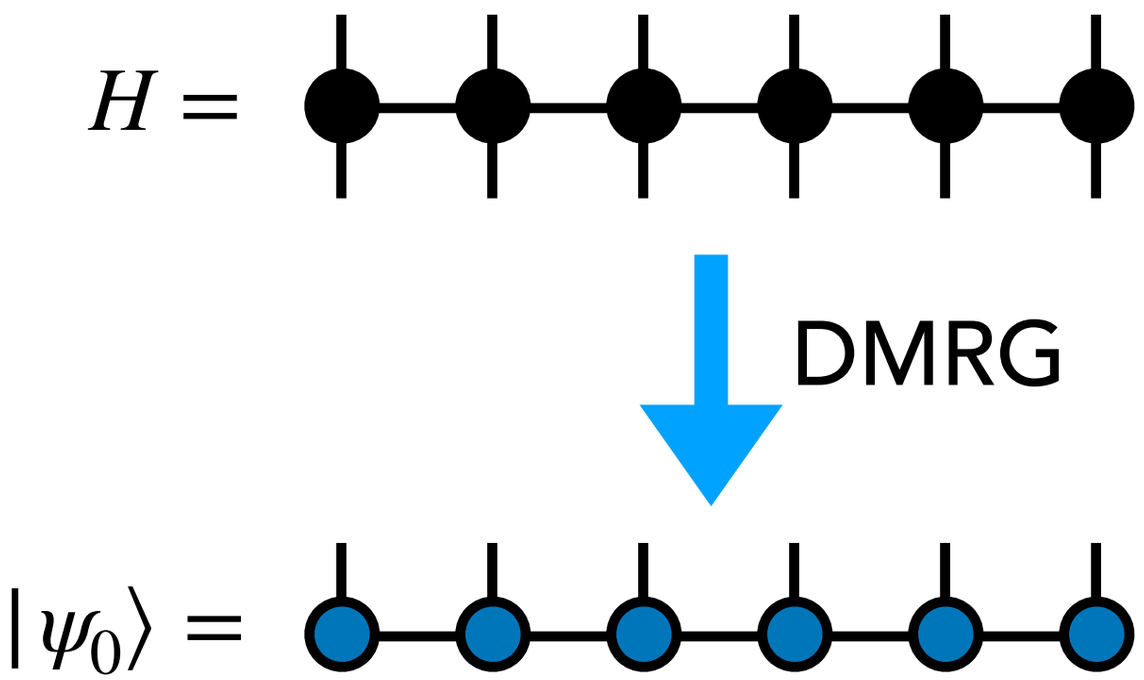
Assume we can write H as a tensor network

$H =$ 

Tensor Network Algorithms

DMRG algorithm

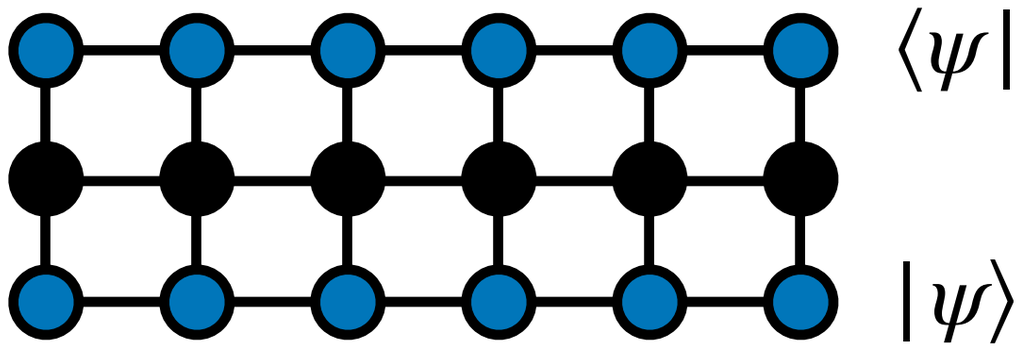
DMRG finds its ground state as an MPS tensor network



Tensor Network Algorithms

DMRG algorithm

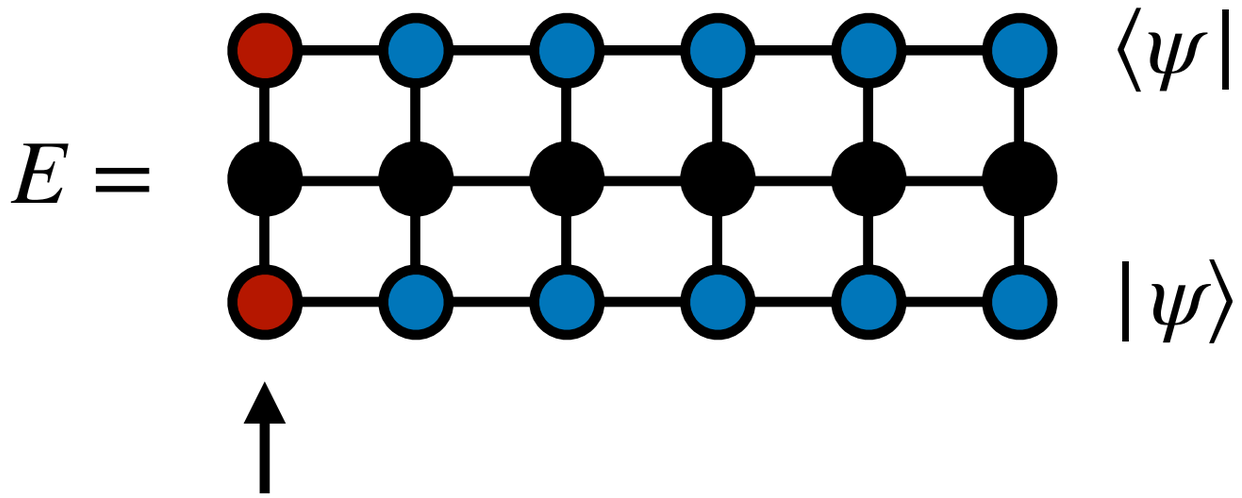
Energy is

$E =$ 

Tensor Network Algorithms

DMRG algorithm

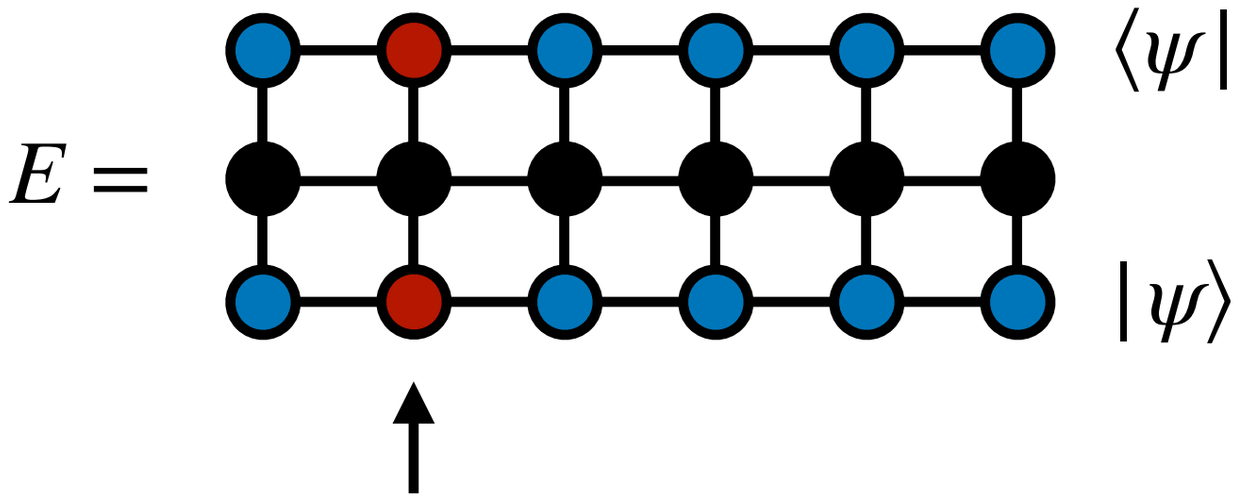
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

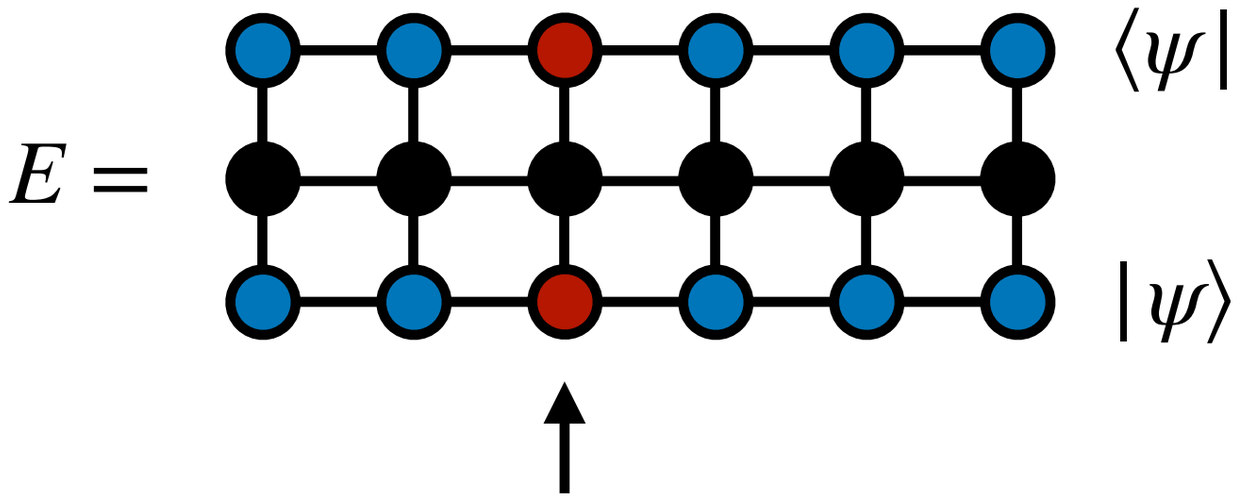
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

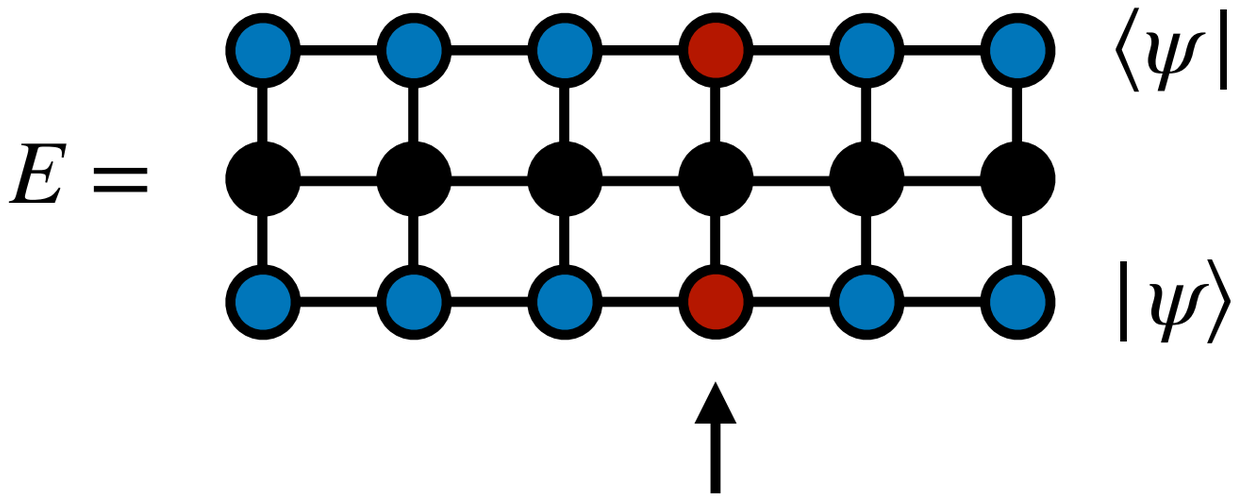
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

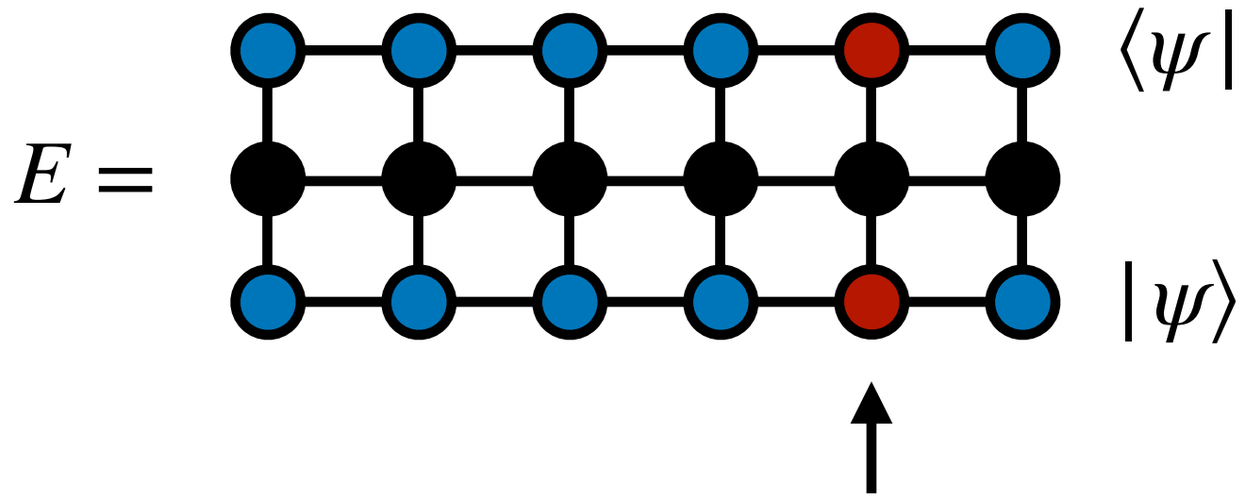
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

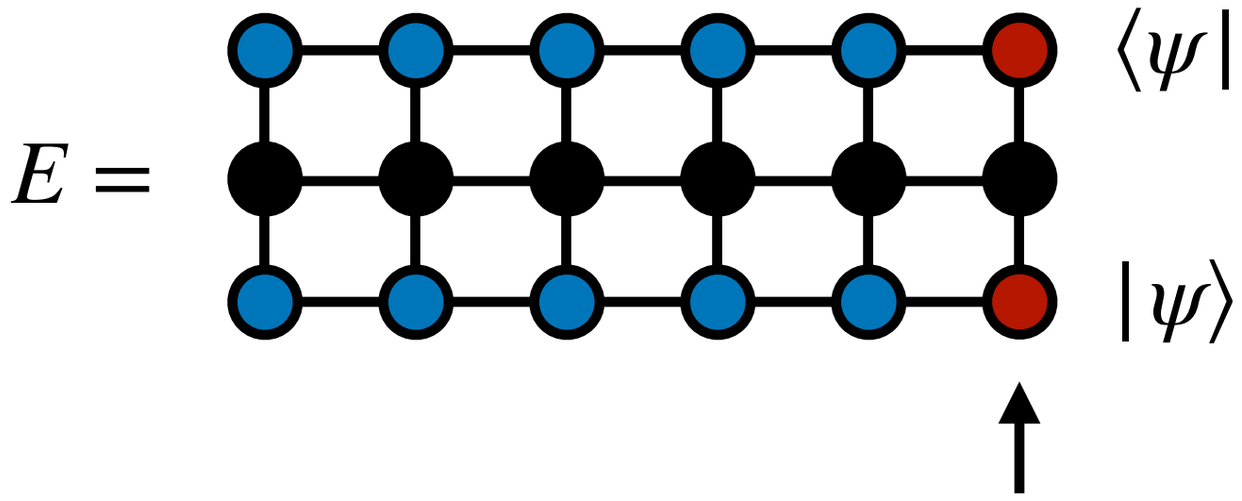
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

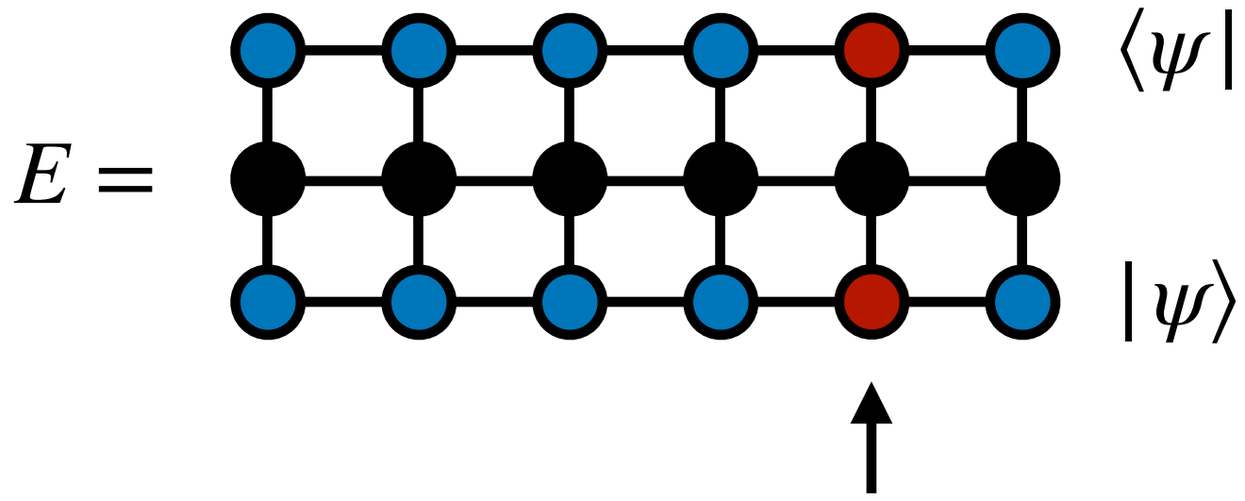
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

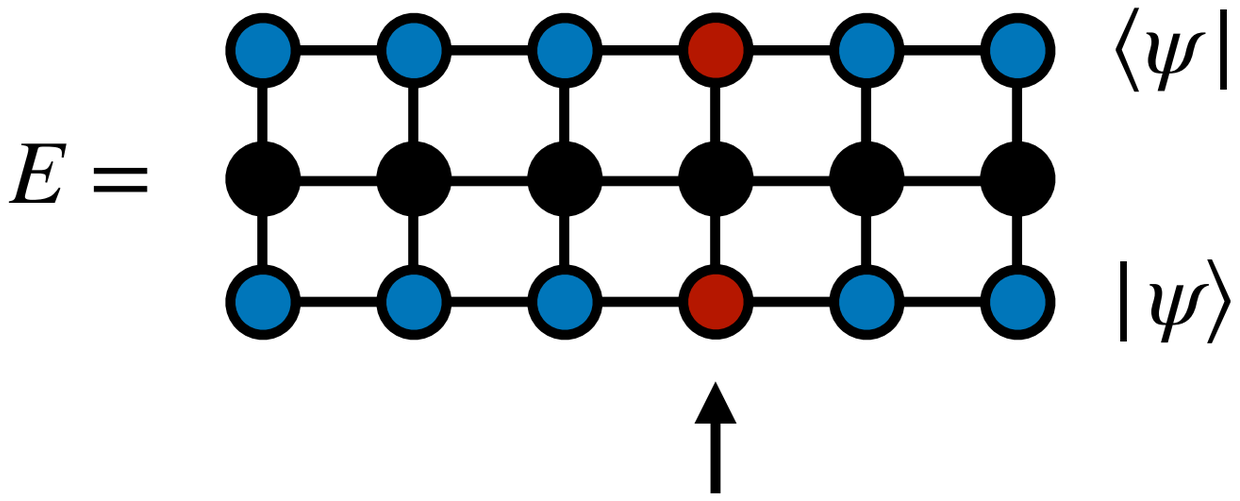
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

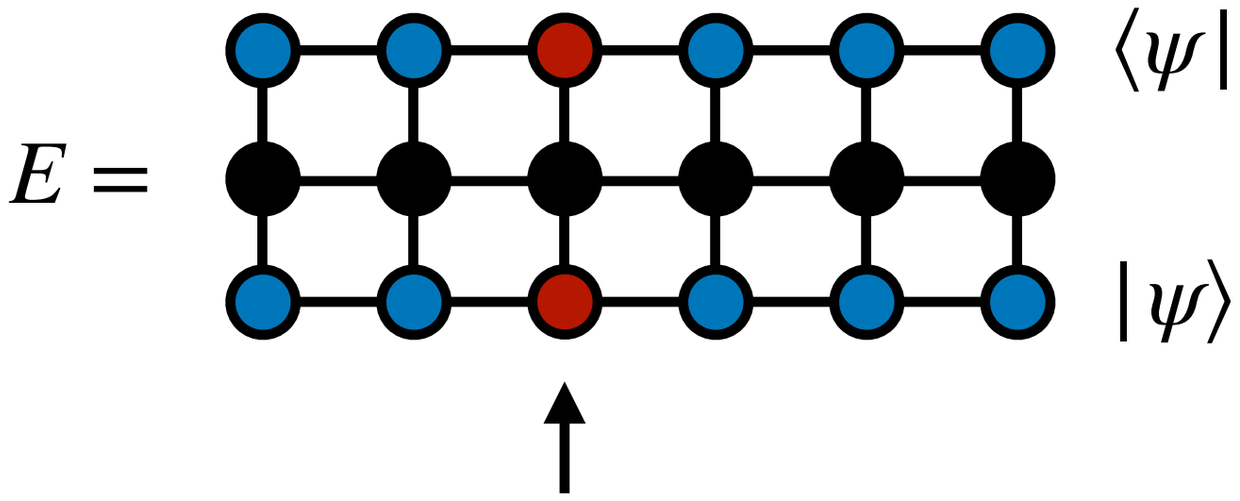
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

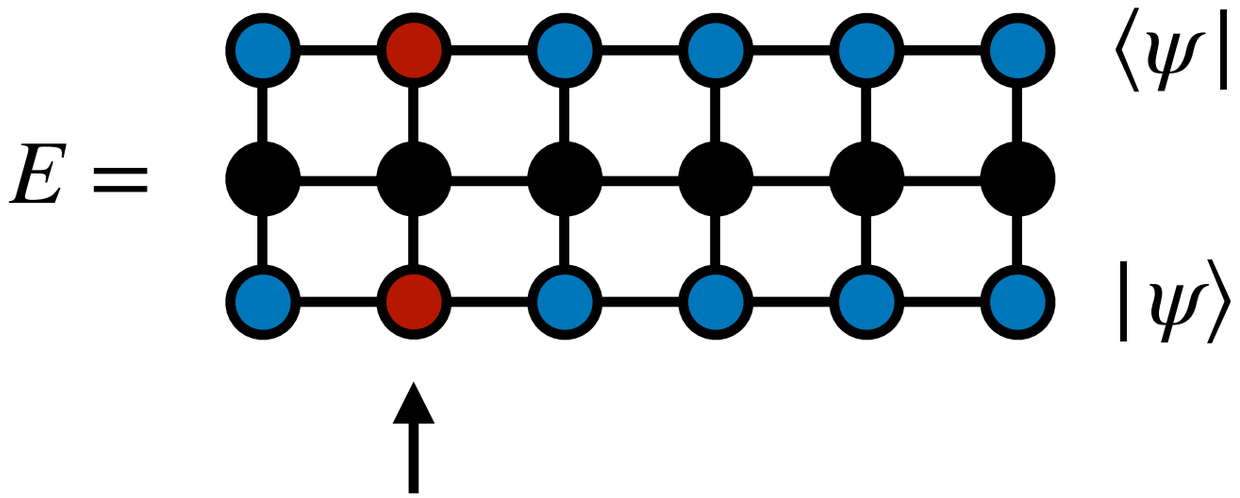
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

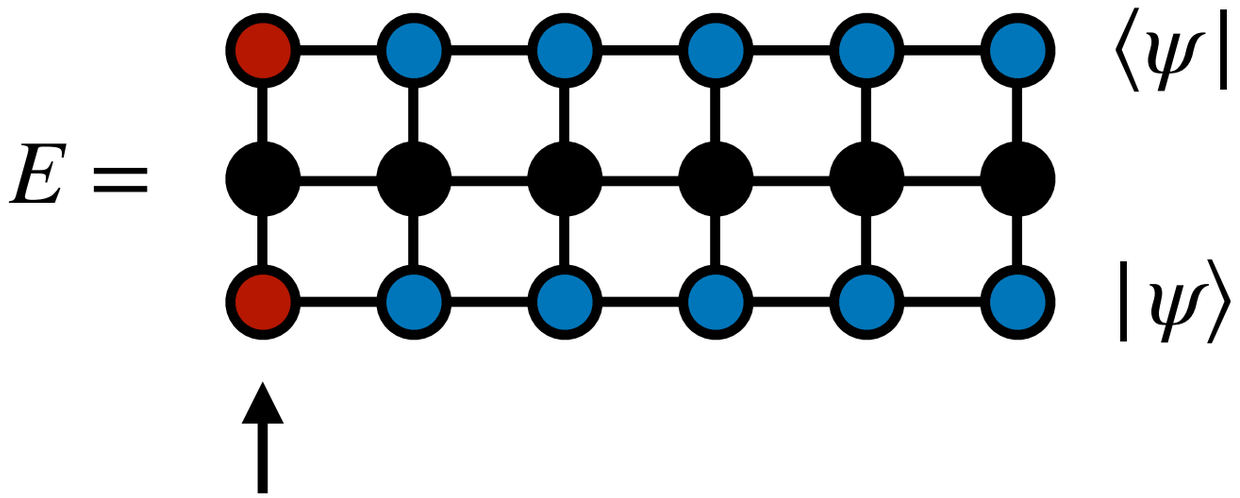
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

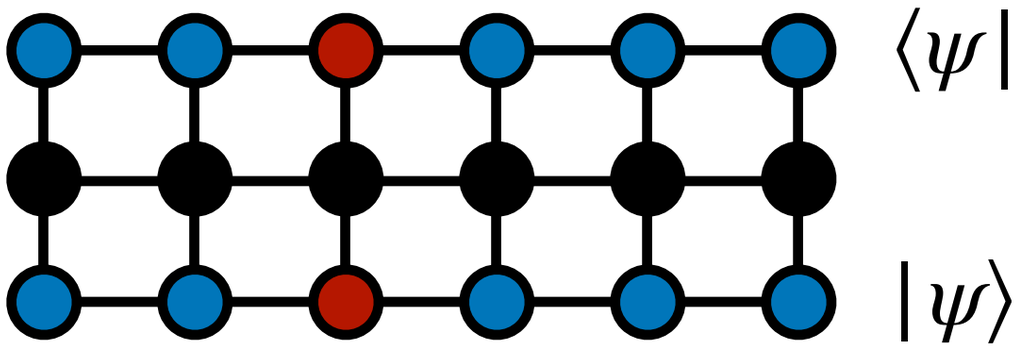
DMRG uses an "alternating" strategy to optimize



Tensor Network Algorithms

DMRG algorithm

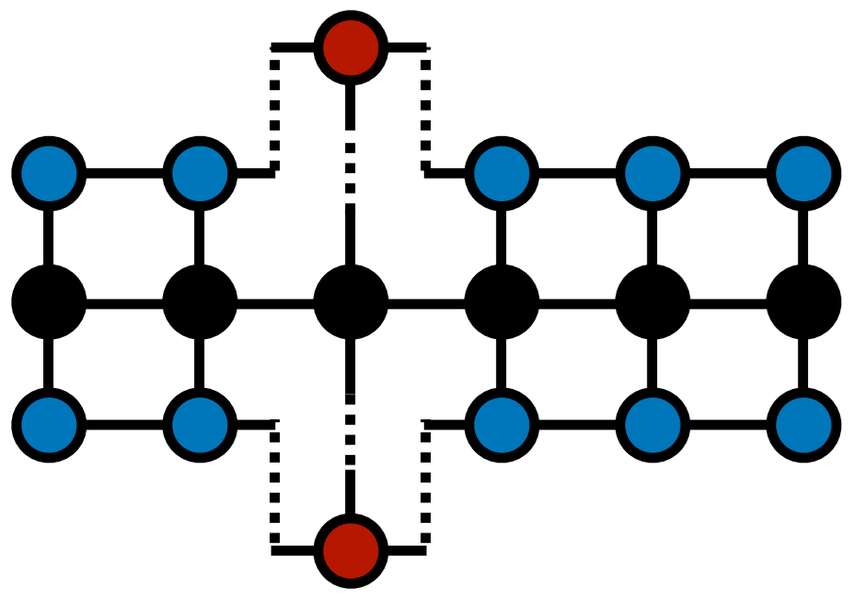
At each step, solve a "mini" diagonalization problem

$E =$ 

Tensor Network Algorithms

DMRG algorithm

At each step, solve a "mini" diagonalization problem*

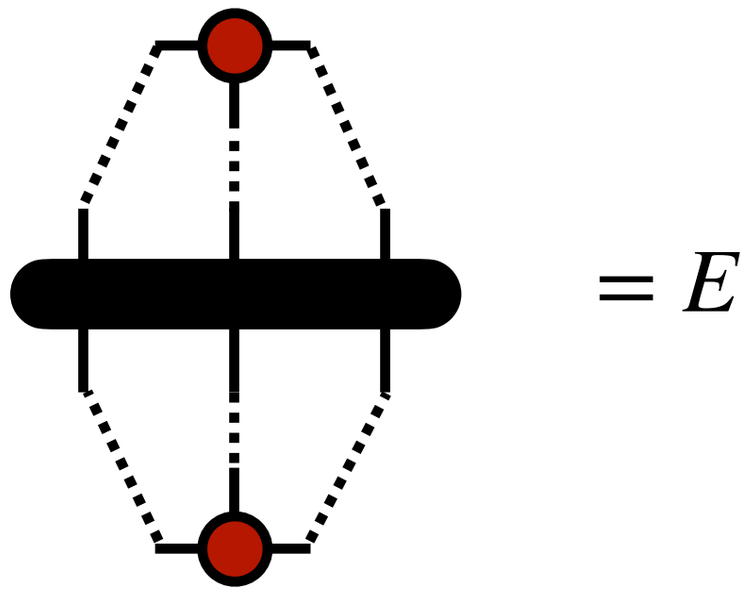


*technical note: for efficiency, frozen tensors are contracted in three groups, not exactly as shown above

Tensor Network Algorithms

DMRG algorithm

At each step, solve a "mini" diagonalization problem*

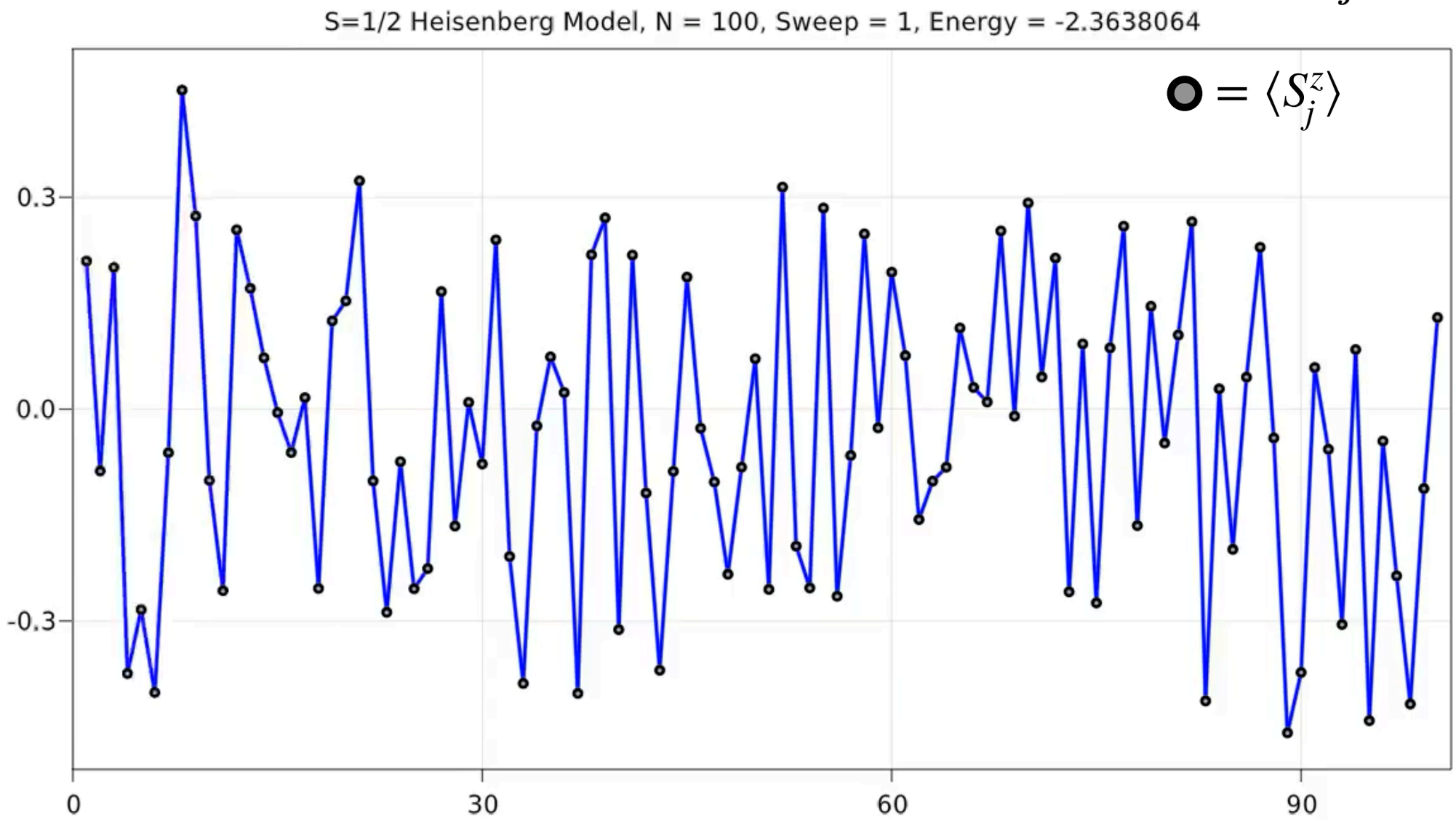


*technical note: for efficiency, frozen tensors are contracted in three groups, not exactly as shown above

Tensor Network Algorithms

DMRG ground-state finding algorithm

DMRG in action – solving Heisenberg chain $H = \sum_j \vec{S}_j \cdot \vec{S}_{j+1}$



Tensor Network Algorithms

DMRG algorithm is extremely precise

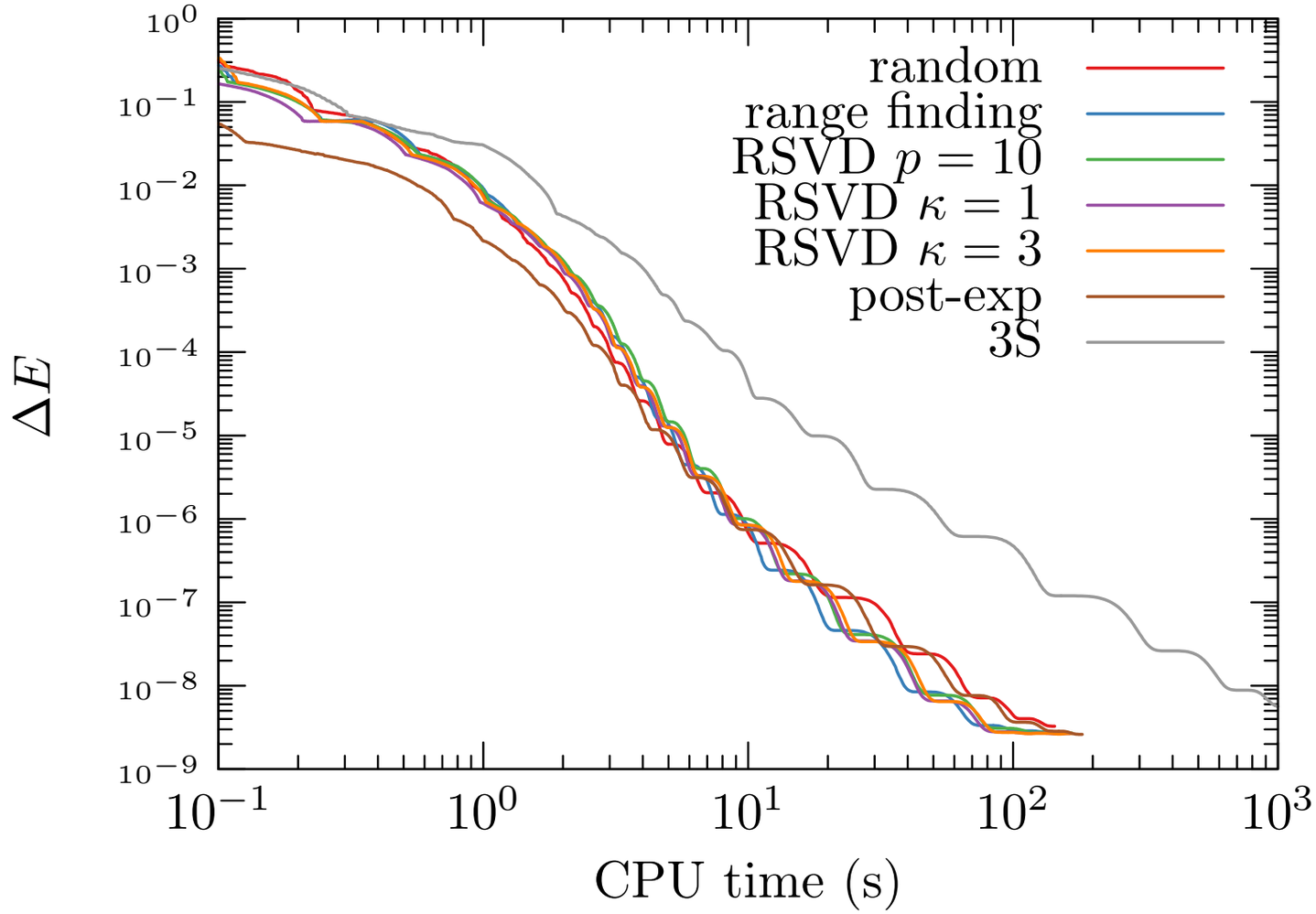
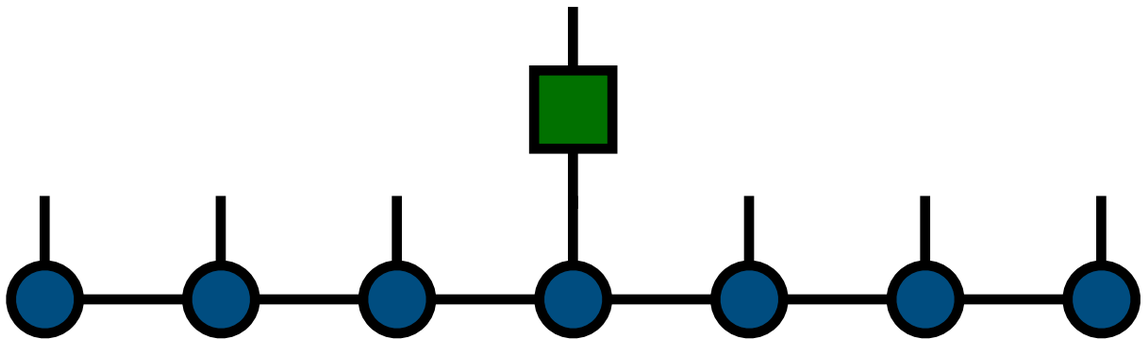


FIG. 5. CPU time (seconds) for the Hubbard-Holstein model.

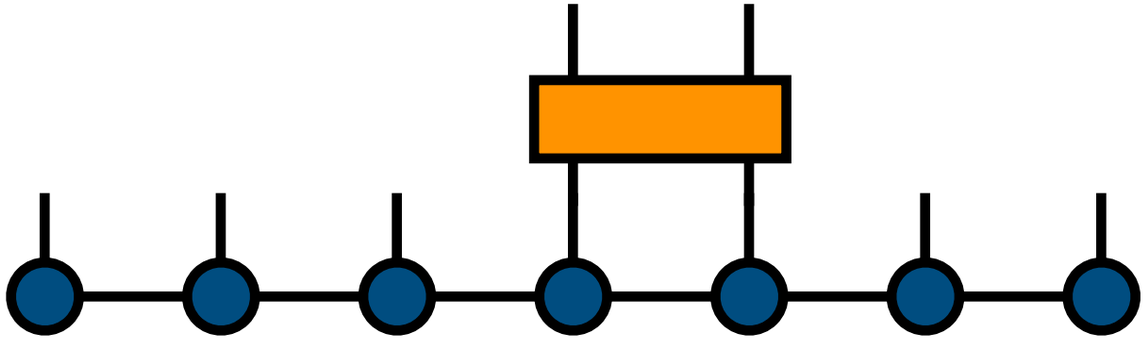
McCulloch, Osborne, arxiv:2403.00562 (2024)

Tensor Network Algorithms

TEBD = controlled application of one-qubit and two-qubit gates to an MPS



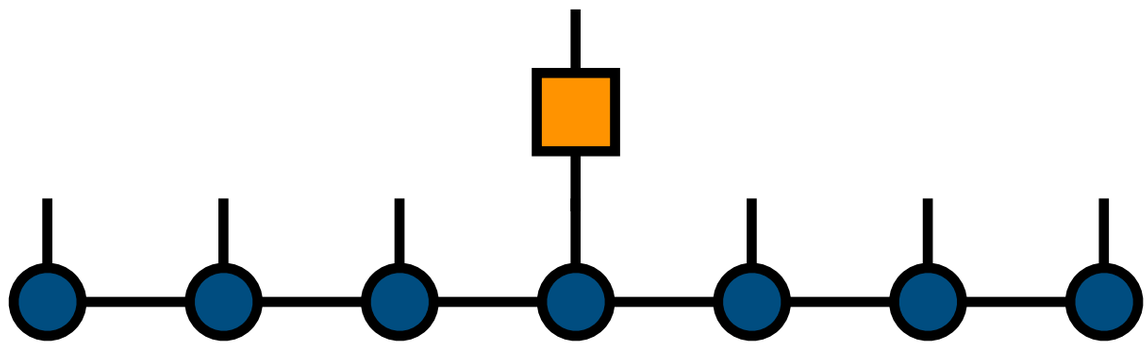
one-qubit gate is exact



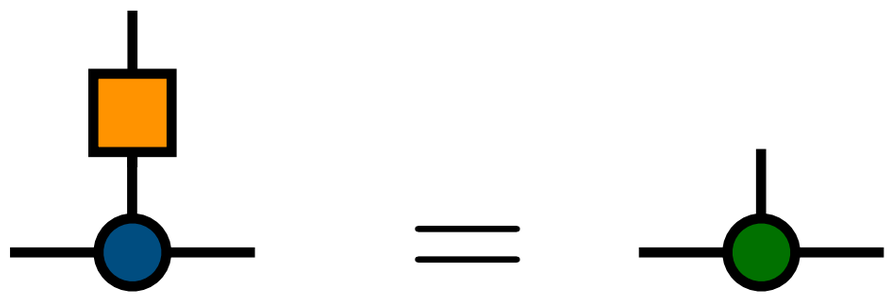
two-qubit gate incurs small *error* controlled by bond-dimension χ

Tensor Network Algorithms

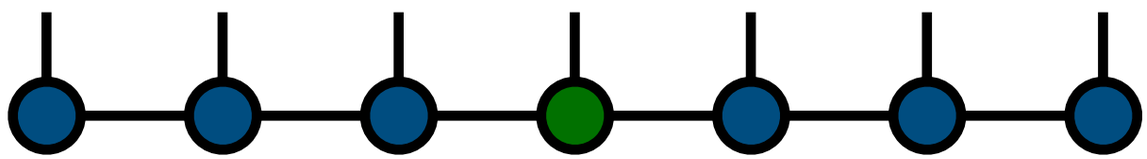
Say we want to act a single-qubit gate on a wavefunction in MPS form



Just operate on one tensor:

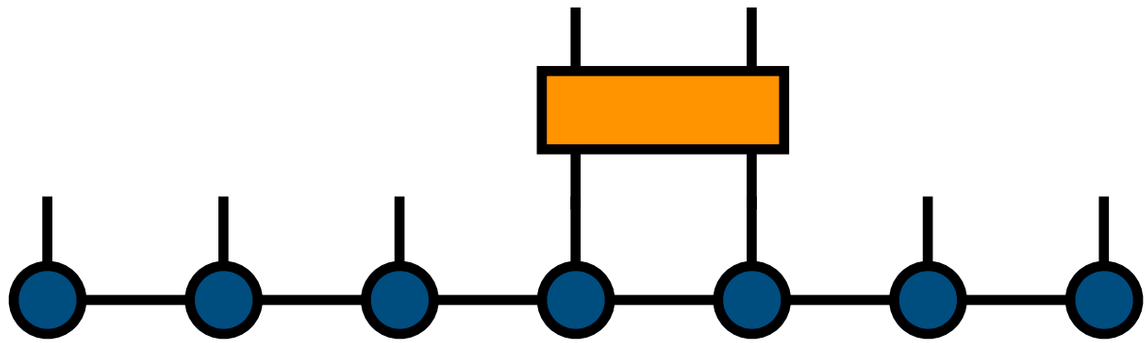


Result:

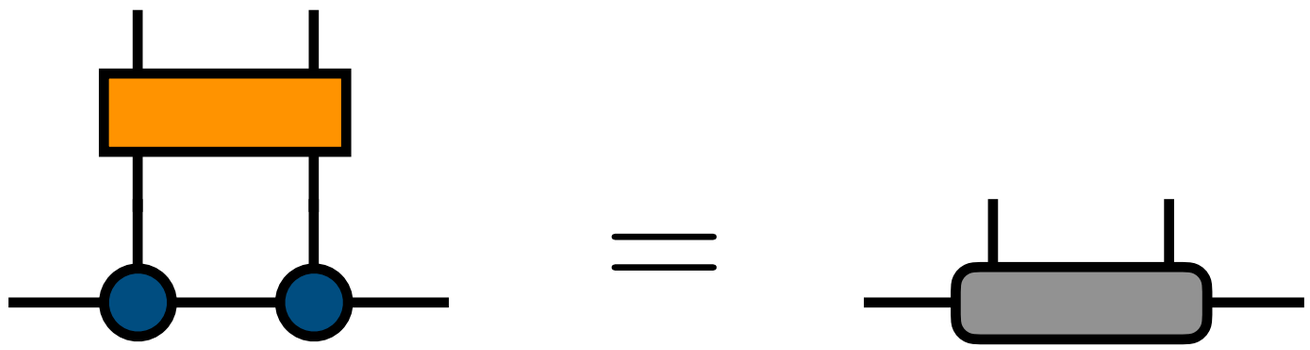


same bond dimension

Say we want to act two-qubit gate on a wavefunction in MPS form

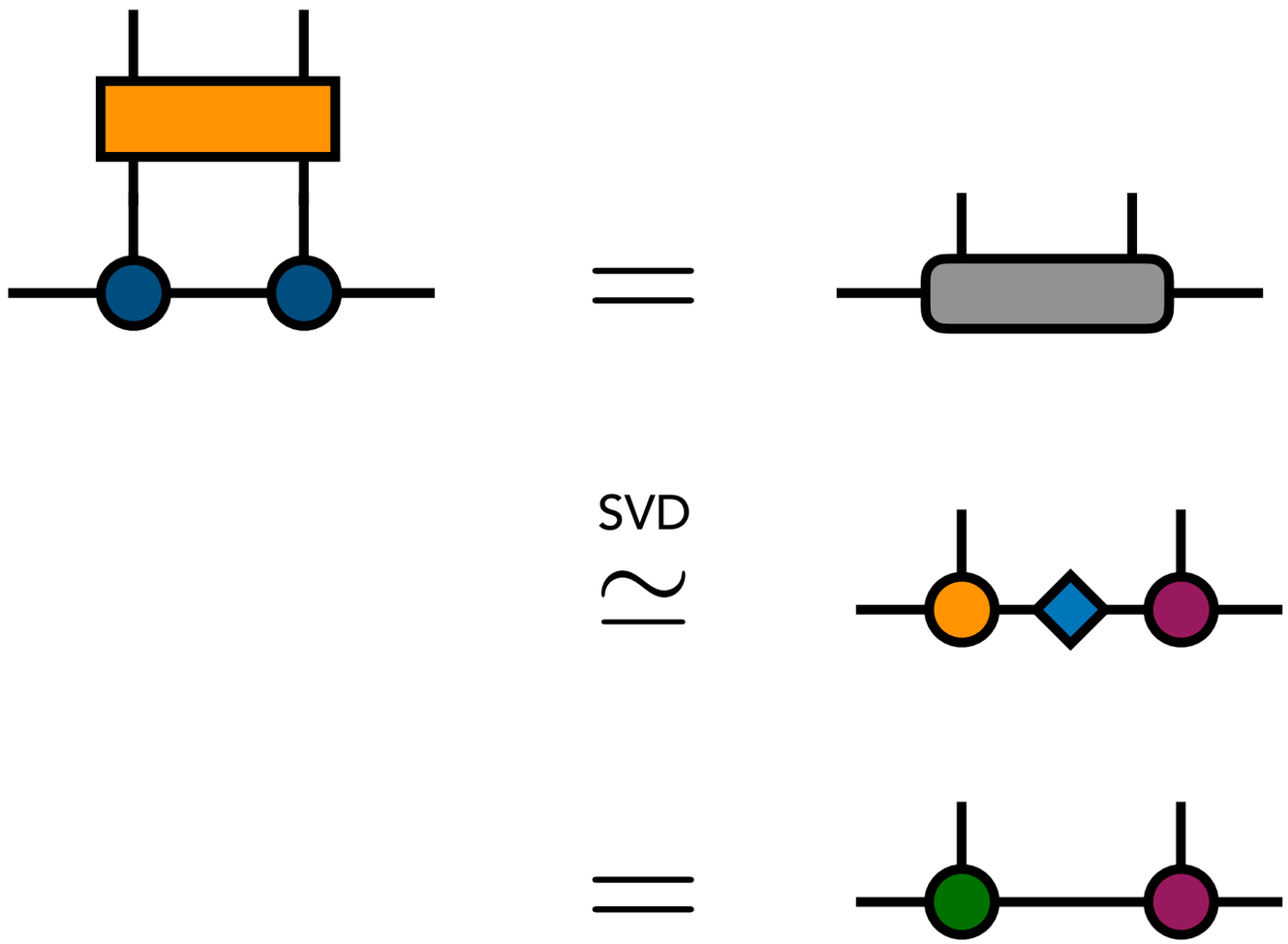


Operate on two tensors:



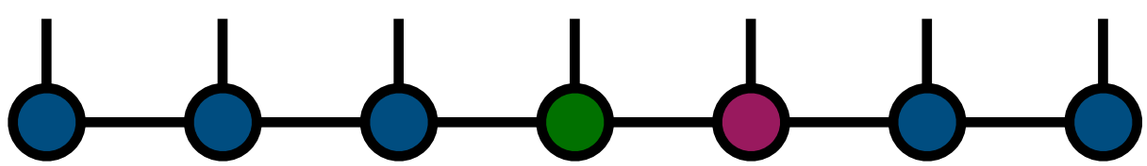
destroy MPS form locally

But can recover MPS form using truncated SVD:



keep top χ *values*

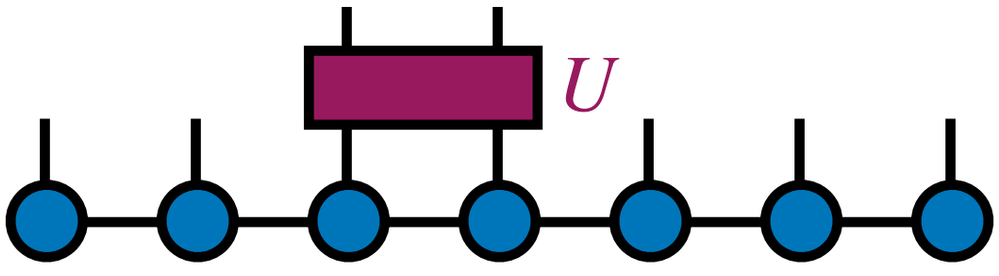
Result:



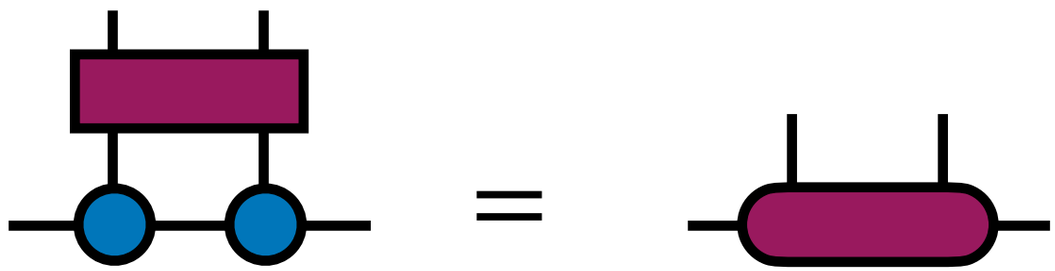
same bond dimension, small loss of fidelity

Tensor Network Algorithms

Summary: applying gates (TEBD algorithm)

$U\,|\Psi\rangle =$ 

1. Contract three tensors



2. Approximately factorize

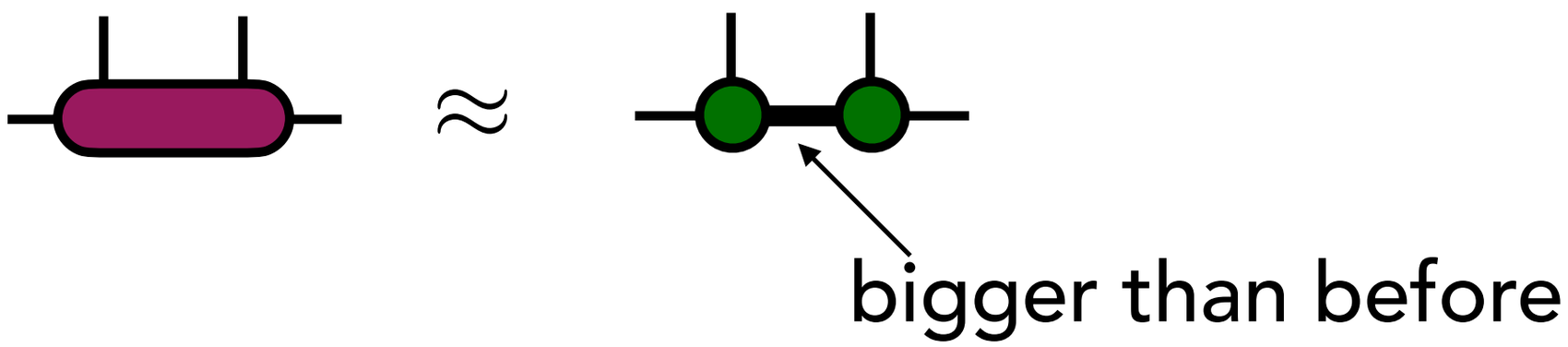
& put back

Tensor Network Algorithms

Applying gates (TEBD algorithm)

Q: What stops us from simulating any quantum circuit?

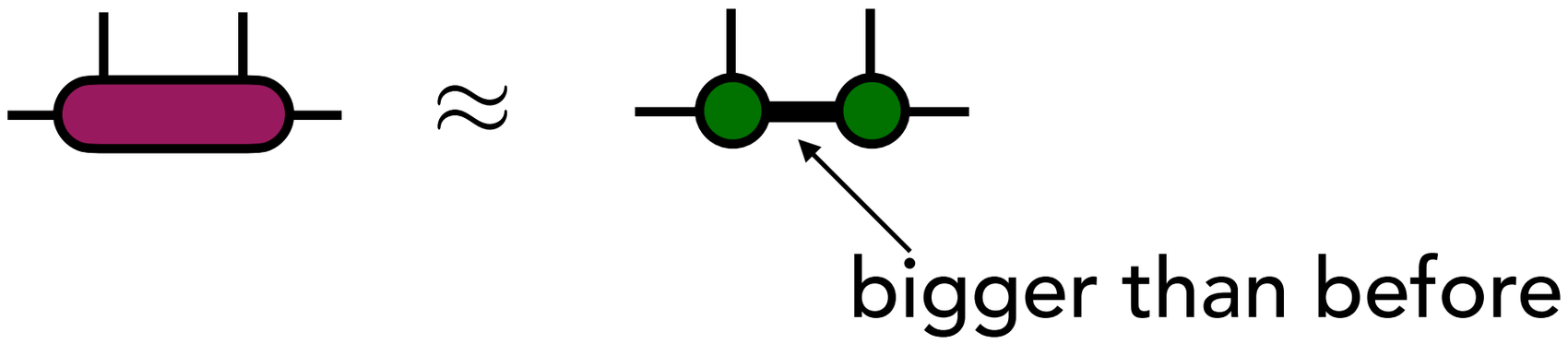
A: Factorization step: internal ranks generally grow to maintain accuracy



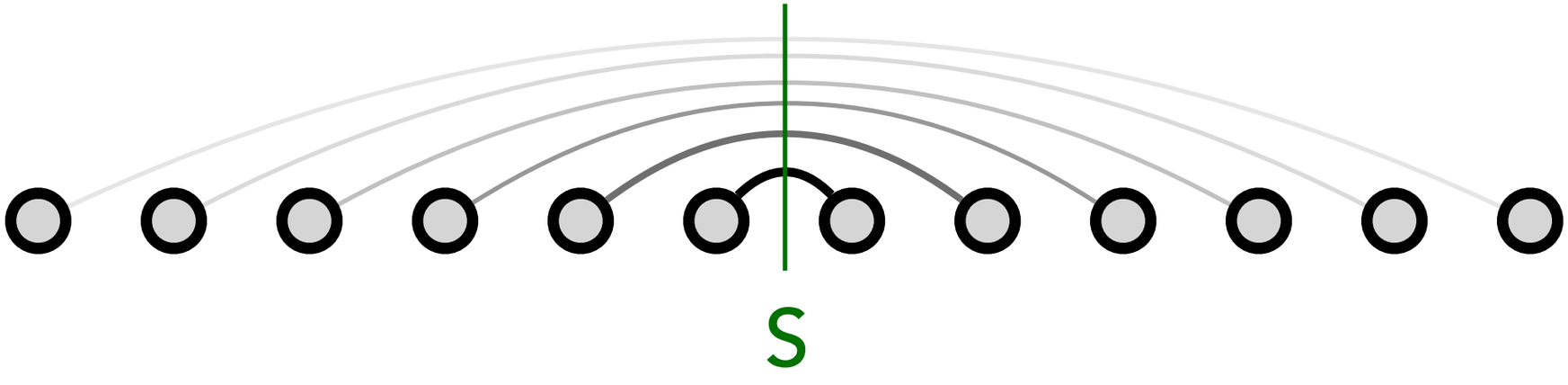
Tensor Network Algorithms

Applying gates (TEBD algorithm)

Q: What stops us from simulating any quantum circuit?



Rank lower bound related to entanglement S of $|\Psi\rangle$ *



* $S \leq \log(\chi)$

Tensor Network Algorithms

Applying gates (TEBD algorithm)

Q: What stops us from simulating any quantum circuit?

Important to note:

- not limited by number of qubits (n > 1000 can be done)
- not limited by type of gates
- agnostic to underlying problem

Entanglement is the barrier! (And relatedly, dimensionality.)

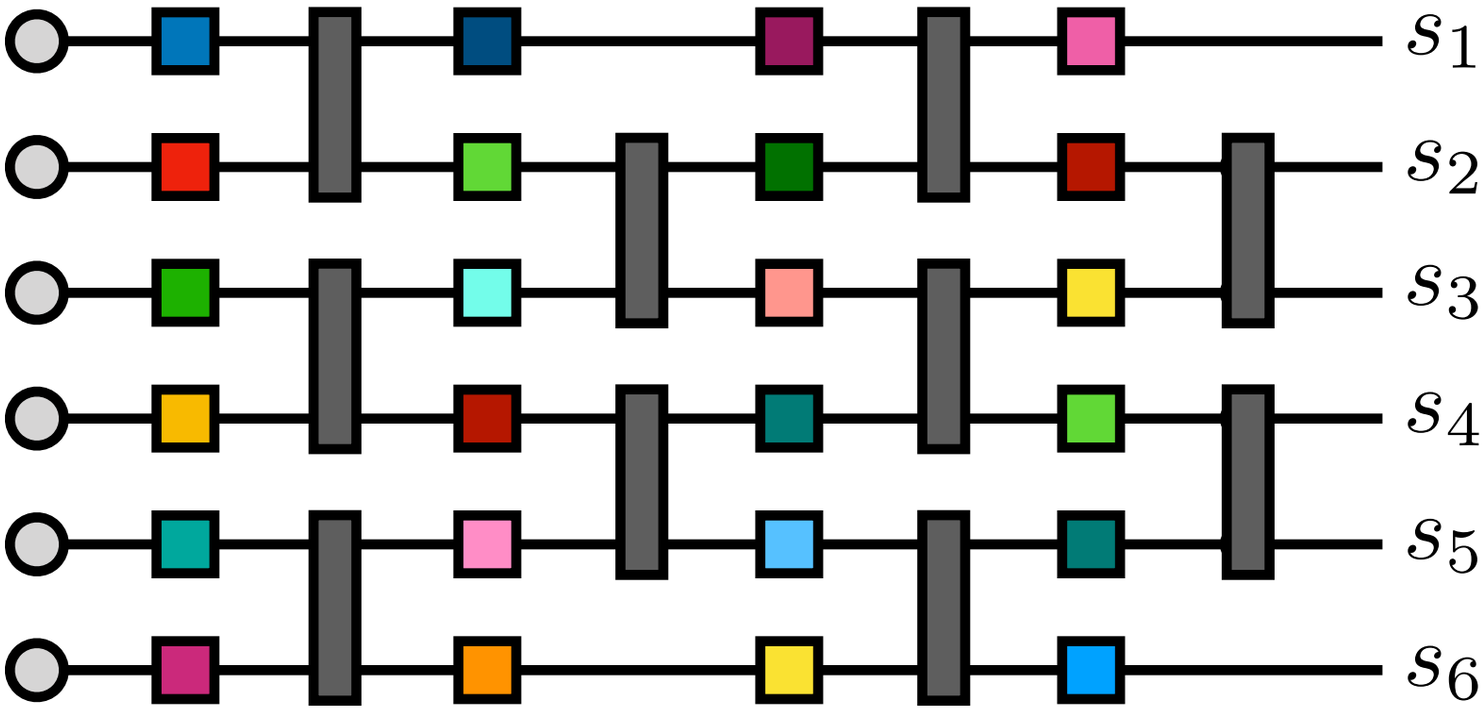
Tensor Network Quantum Simulations

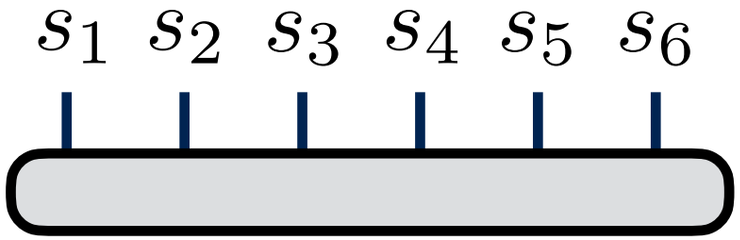
References:

- Zhou, EMS, Waintal, "What Limits the Simulation of Quantum Computers?", PRX 10, 041038 (2020)
- Ayral et al., "DMRG Algorithm for Simulating Quantum Circuits with a Finite Fidelity", PRX Quantum 4, 020304 (2023)
- Vidal, "Efficient Classical Simulation of Slightly Entangled Quantum Computations", Phys. Rev. Lett. 91, 147902 (2003)
- Markov, Shi, "Simulating Quantum Computation by Contracting Tensor Networks", SIAM J. Computing, 38, 963-981 (2001)

Quantum Circuits

After enough gates, state can get complicated

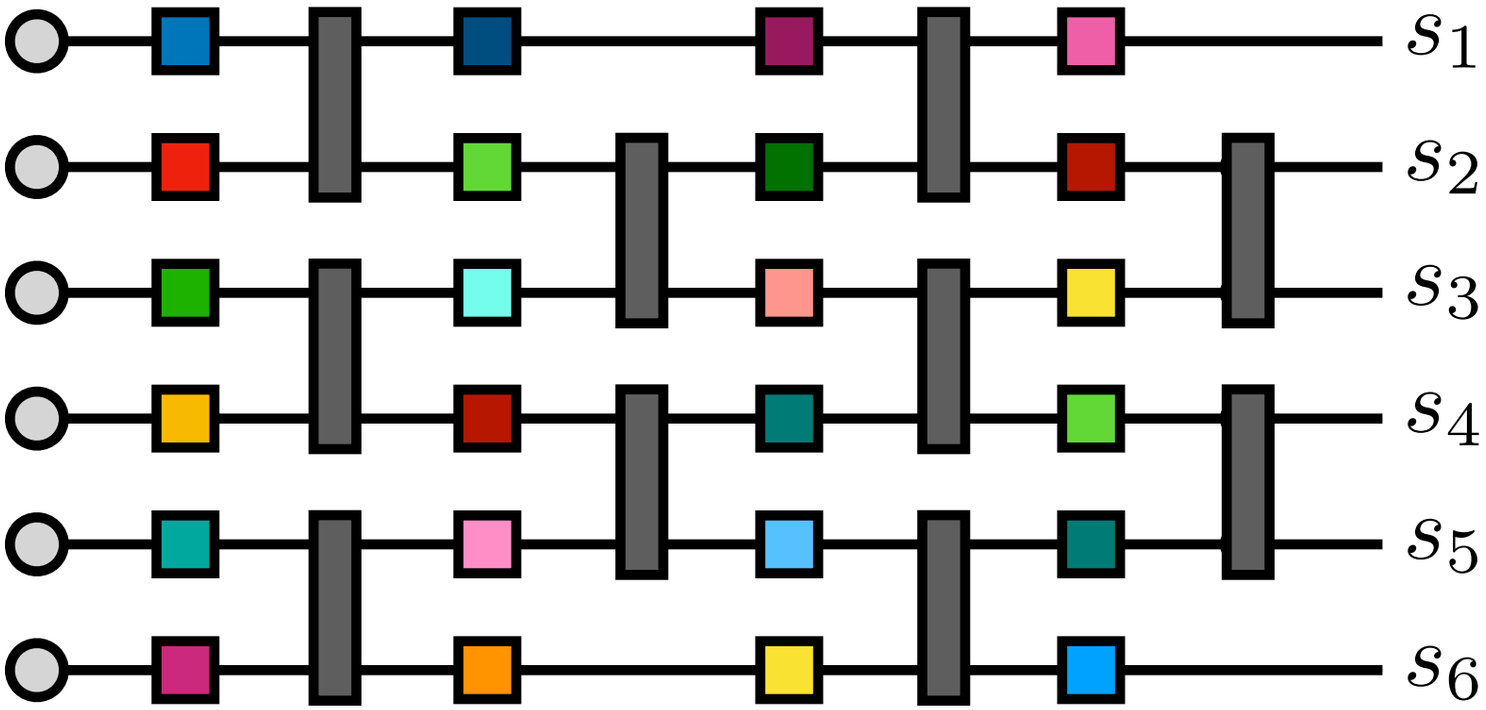


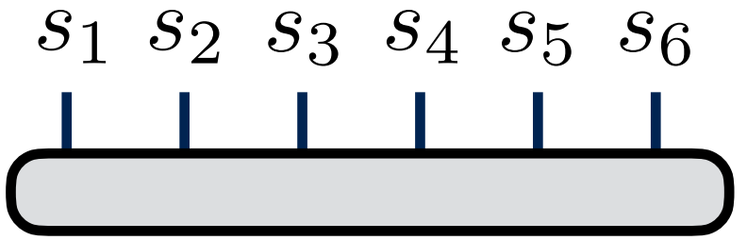
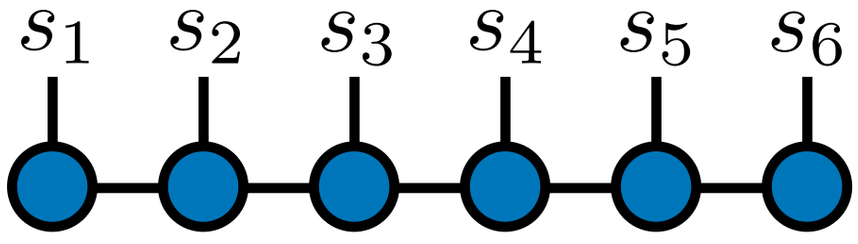
$|\Psi\rangle =$ 

complicated tensor defined by circuit (quantum state)

Quantum Circuits

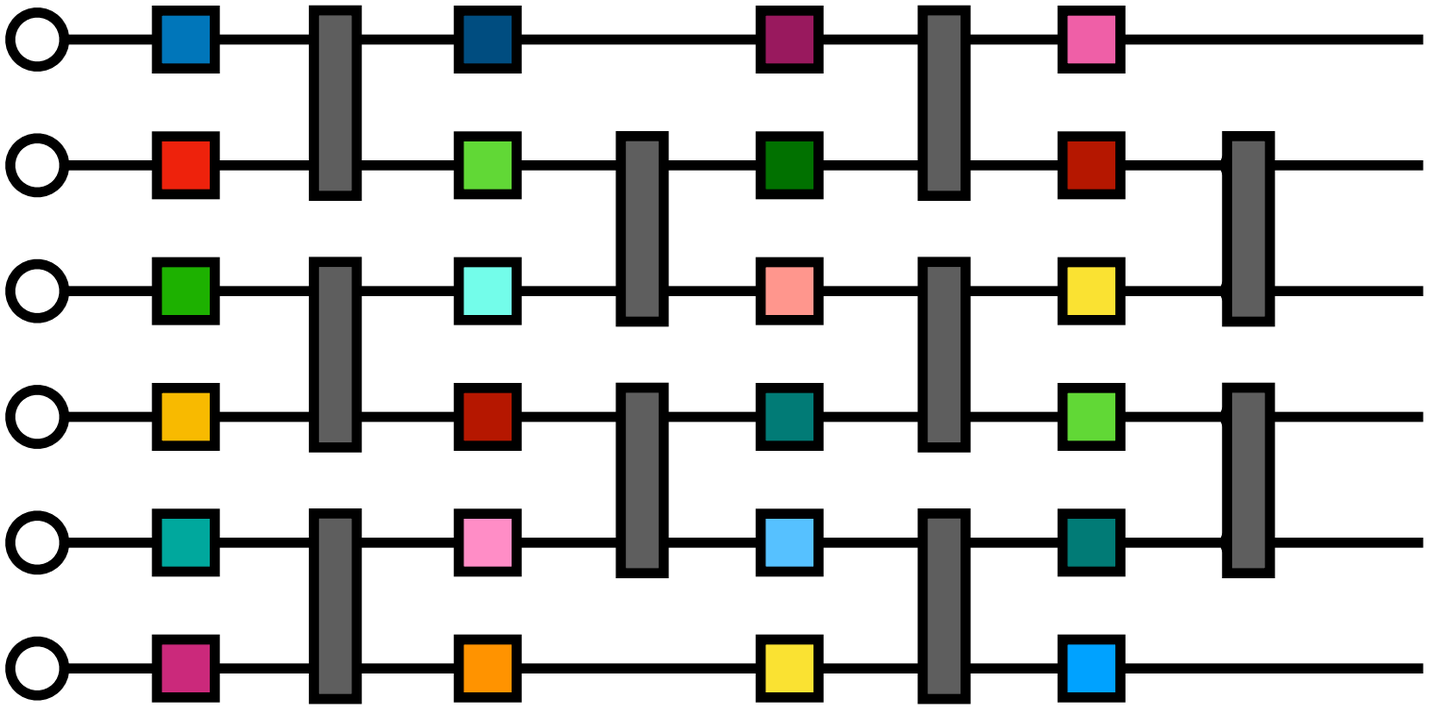
For structured circuits, final state *might* be MPS



$|\Psi\rangle =$  $\approx$ 

TEBD Algorithm

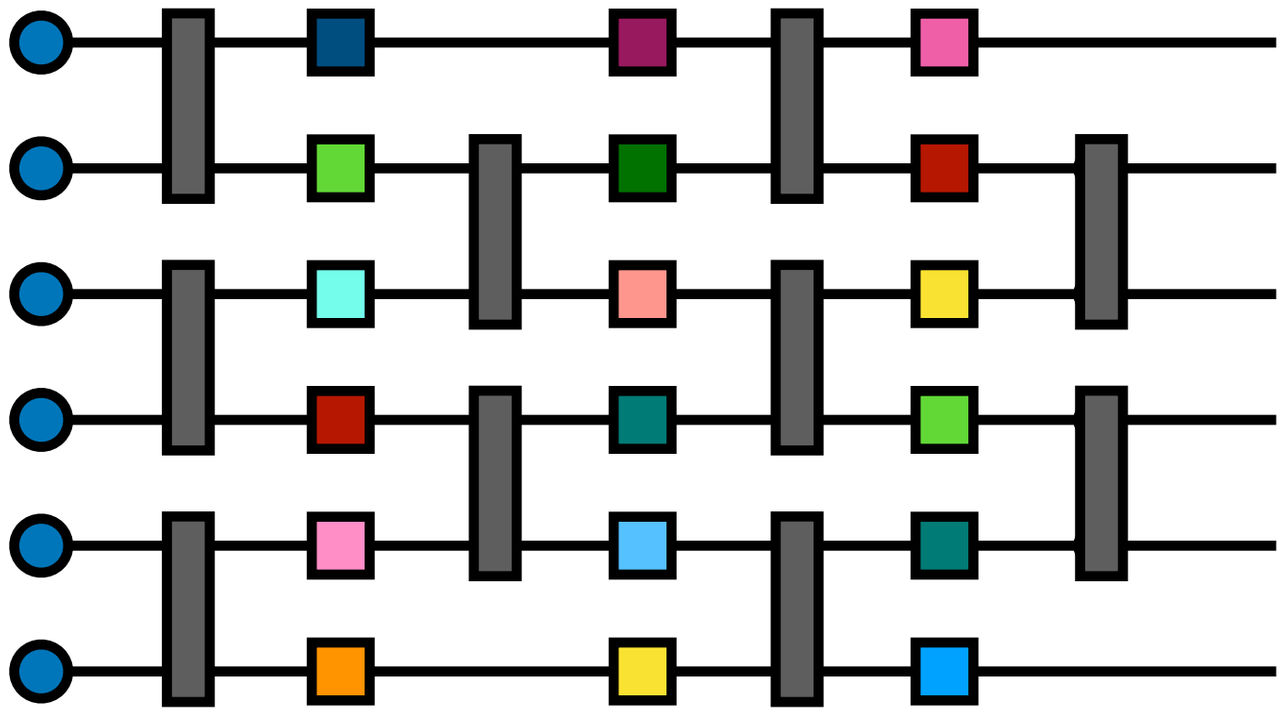
Simulation of 1D quantum circuit with TEBD algorithm



Zhou, EMS, Waintal, "What Limits the Simulation of Quantum Computers?", PRX 10, 041038 (2020)

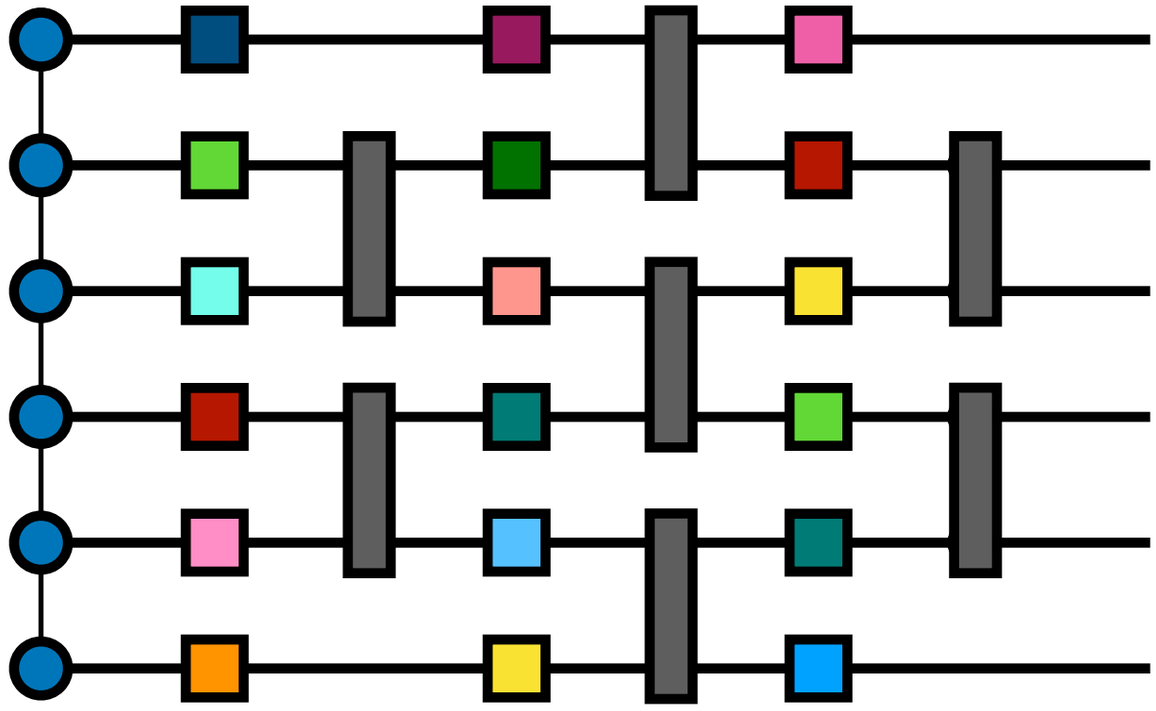
TEBD Algorithm

Simulation of 1D quantum circuit with TEBD algorithm



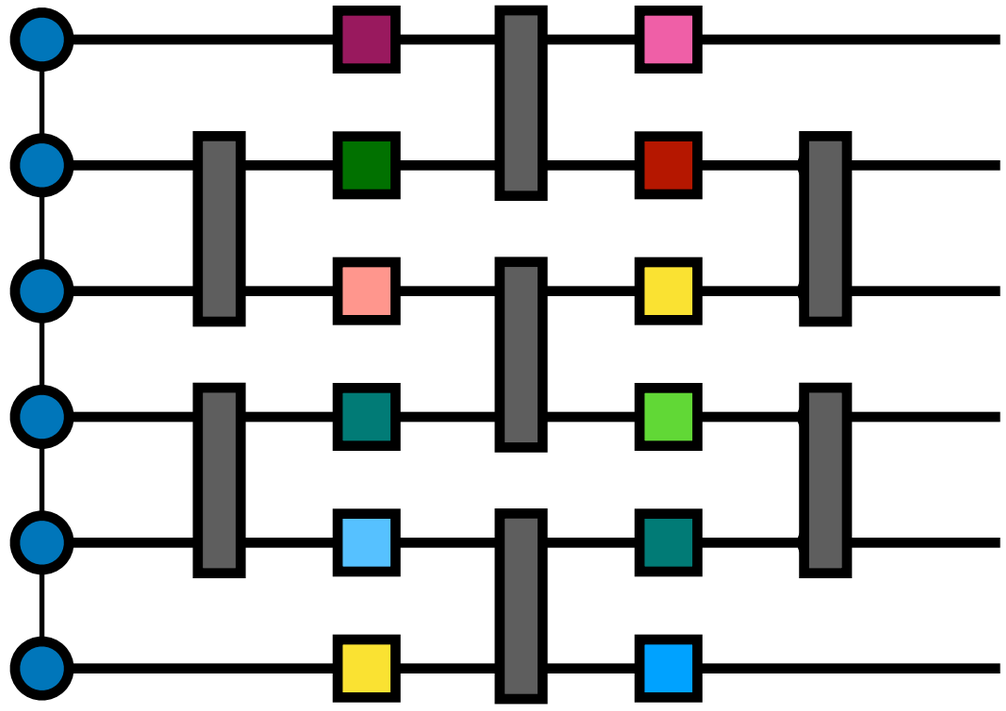
TEBD Algorithm

Simulation of 1D quantum circuit with TEBD algorithm



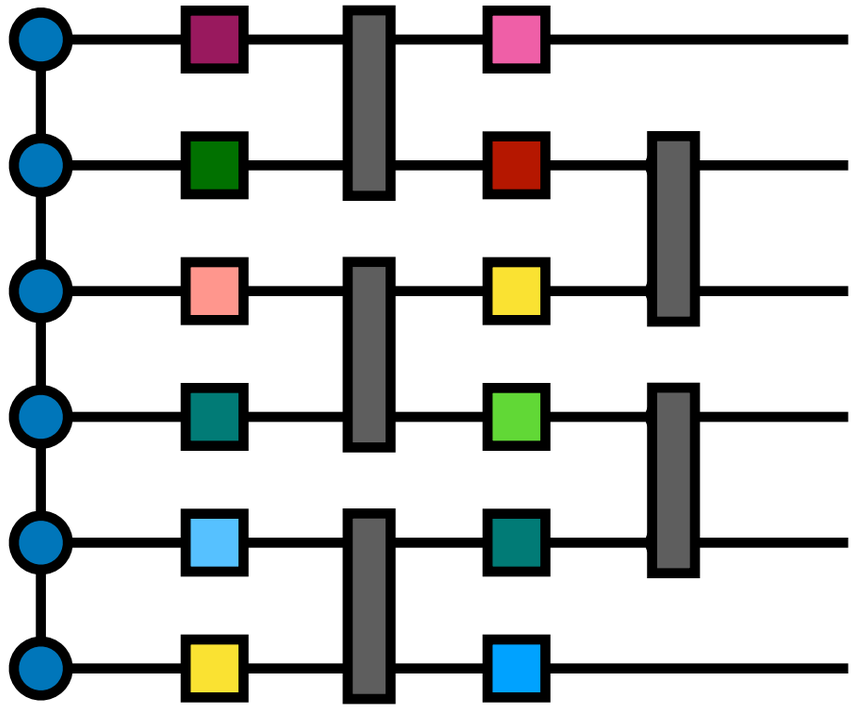
TEBD Algorithm

Simulation of 1D quantum circuit with TEBD algorithm



TEBD Algorithm

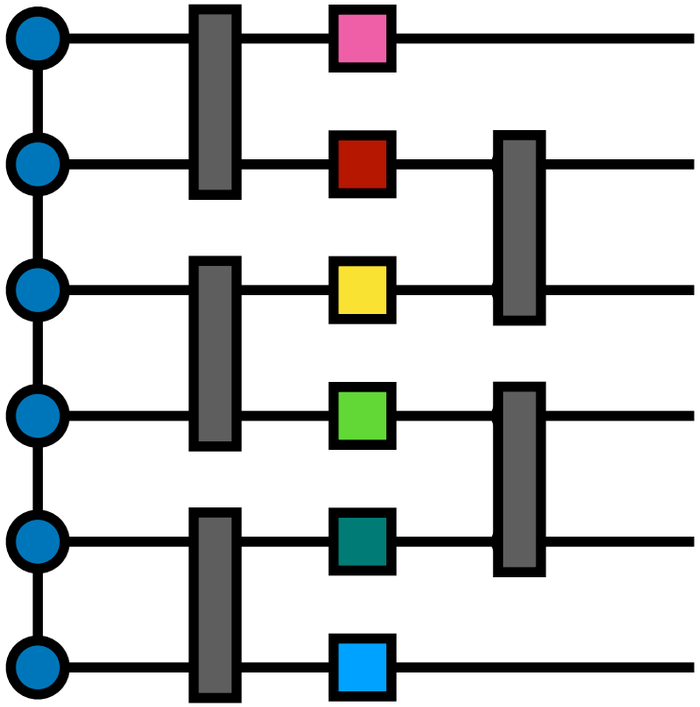
Simulation of 1D quantum circuit with TEBD algorithm



Fidelity

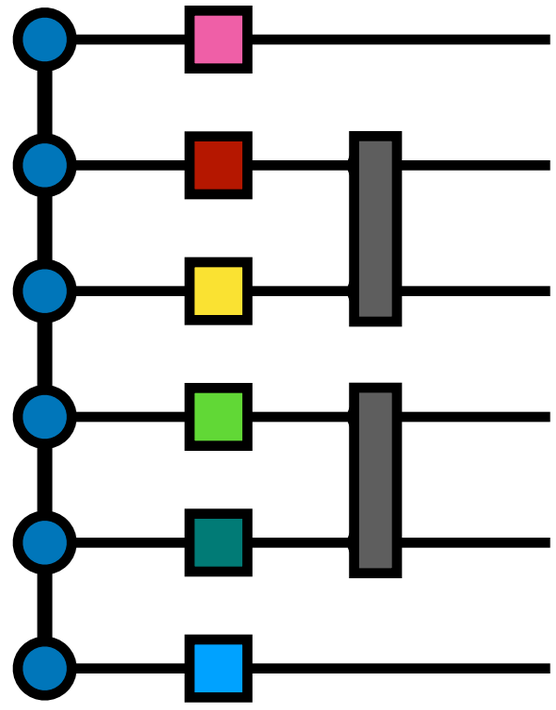
TEBD Algorithm

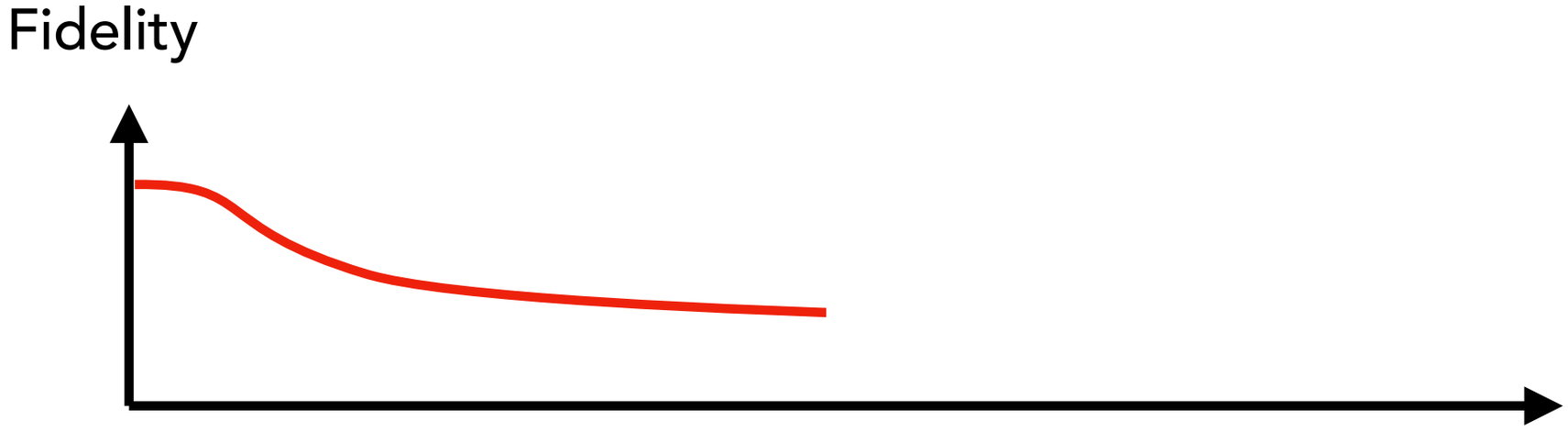
Simulation of 1D quantum circuit with TEBD algorithm



TEBD Algorithm

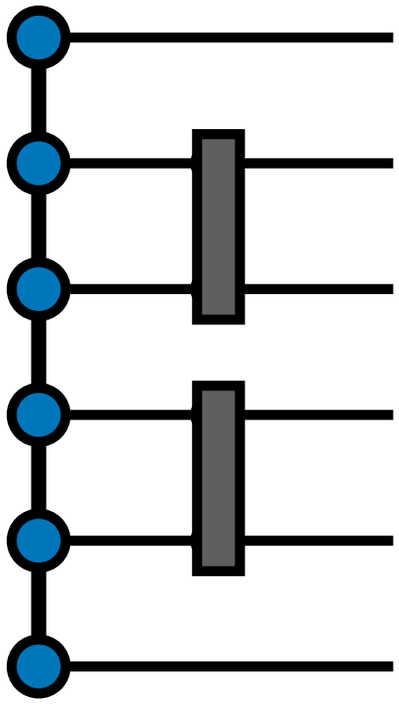
Simulation of 1D quantum circuit with TEBD algorithm

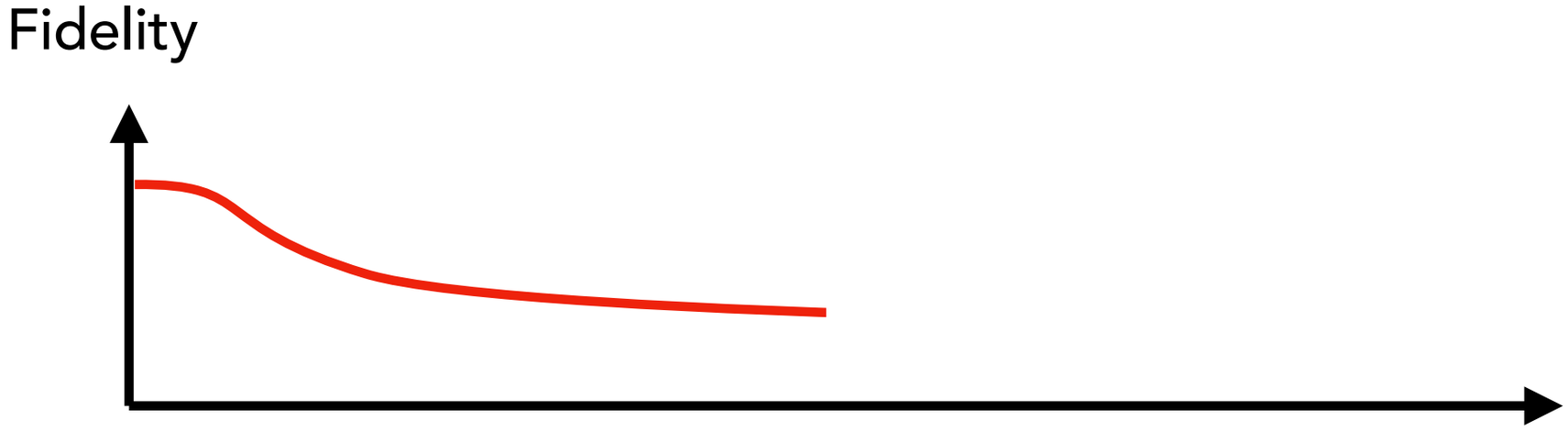




TEBD Algorithm

Simulation of 1D quantum circuit with TEBD algorithm

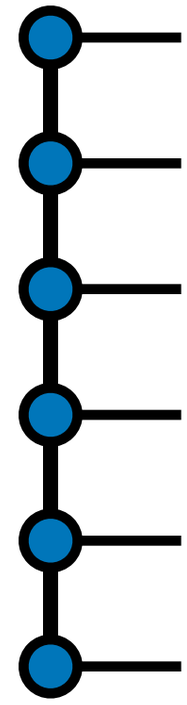


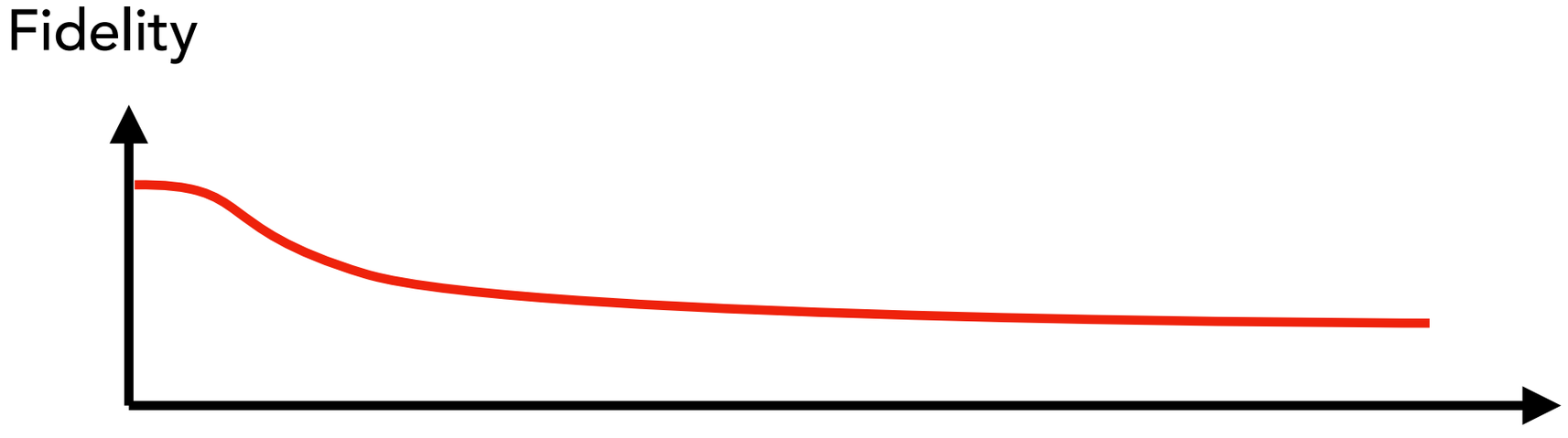


TEBD Algorithm

Simulation of 1D quantum circuit with TEBD algorithm

obtain entire state afterward

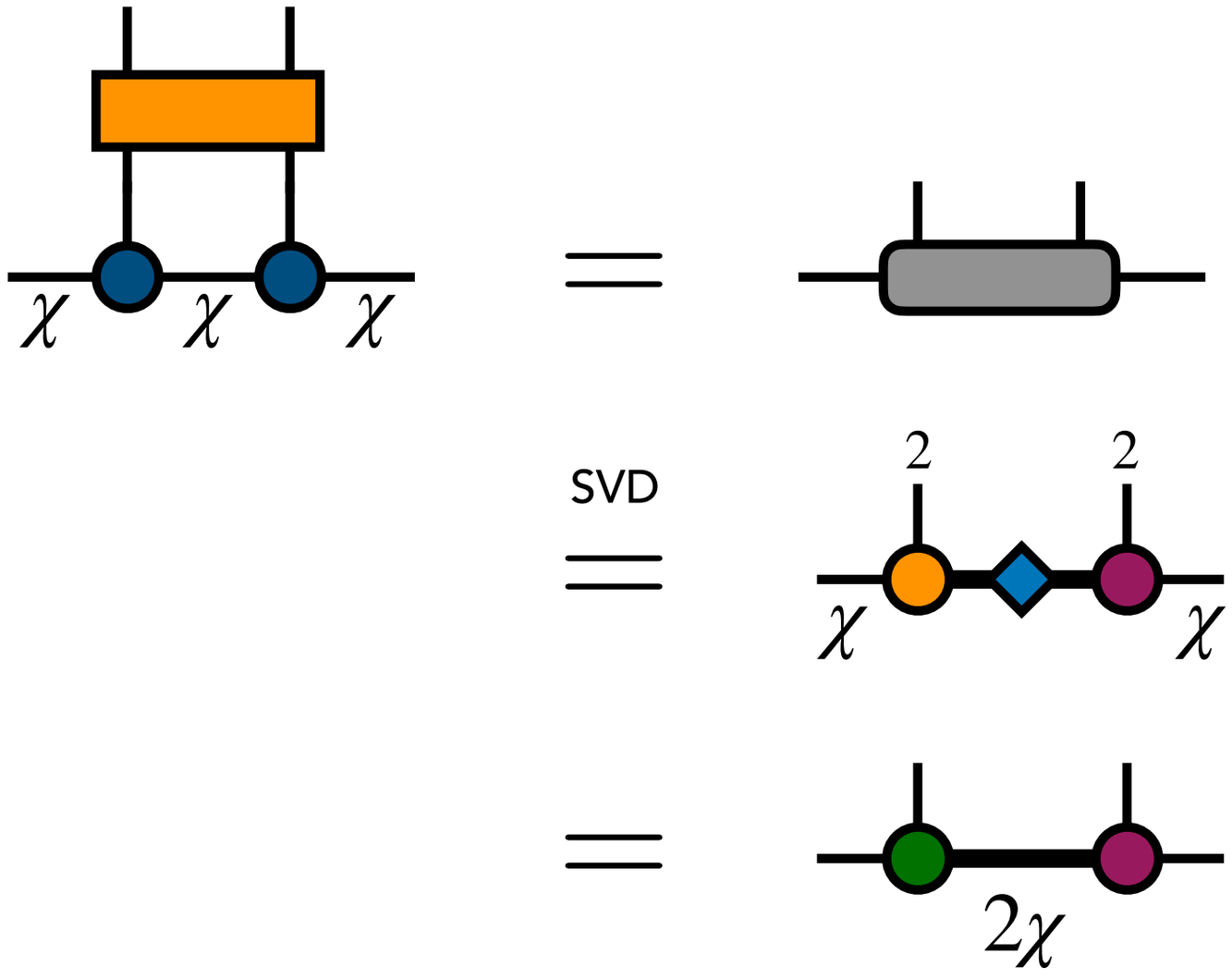




Tensor Network Simulations

When can tensor network simulations succeed?

Worst case, bond dimension χ doubles every layer



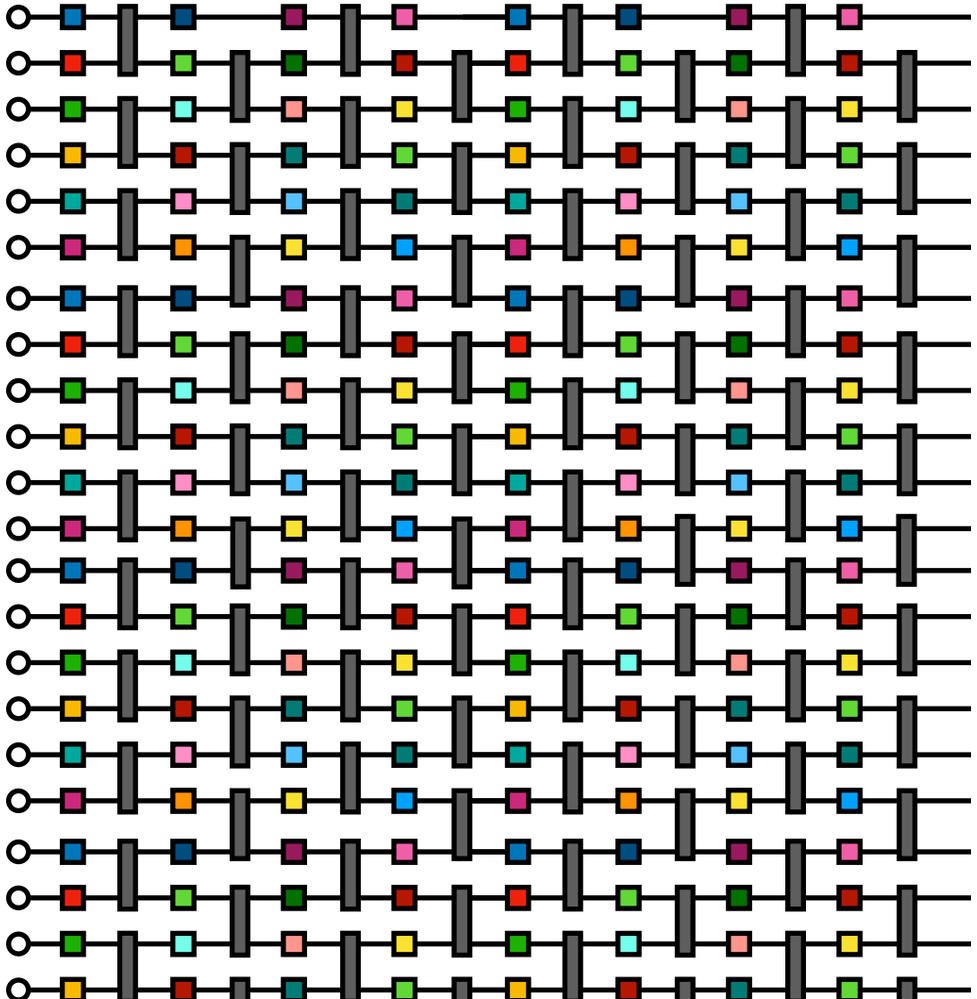
Tensor Network Simulations

Worst case, bond dimension χ doubles every layer

Exponential growth... but
automatically means ~10 layers is always possible in 1D
($2^{10} = 1024$)

Holds even for thousands of qubits

An easy simulation:

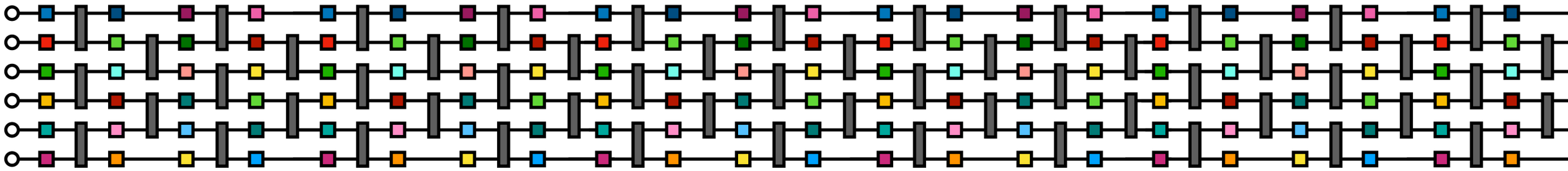


Tensor Network Simulations

Worst case, bond dimension χ doubles every layer

If number of qubits small MPS can also do any number of layers

(20 qubits gives max $\chi = 2^{10}$)



Tensor Network Simulations

Putting this together*

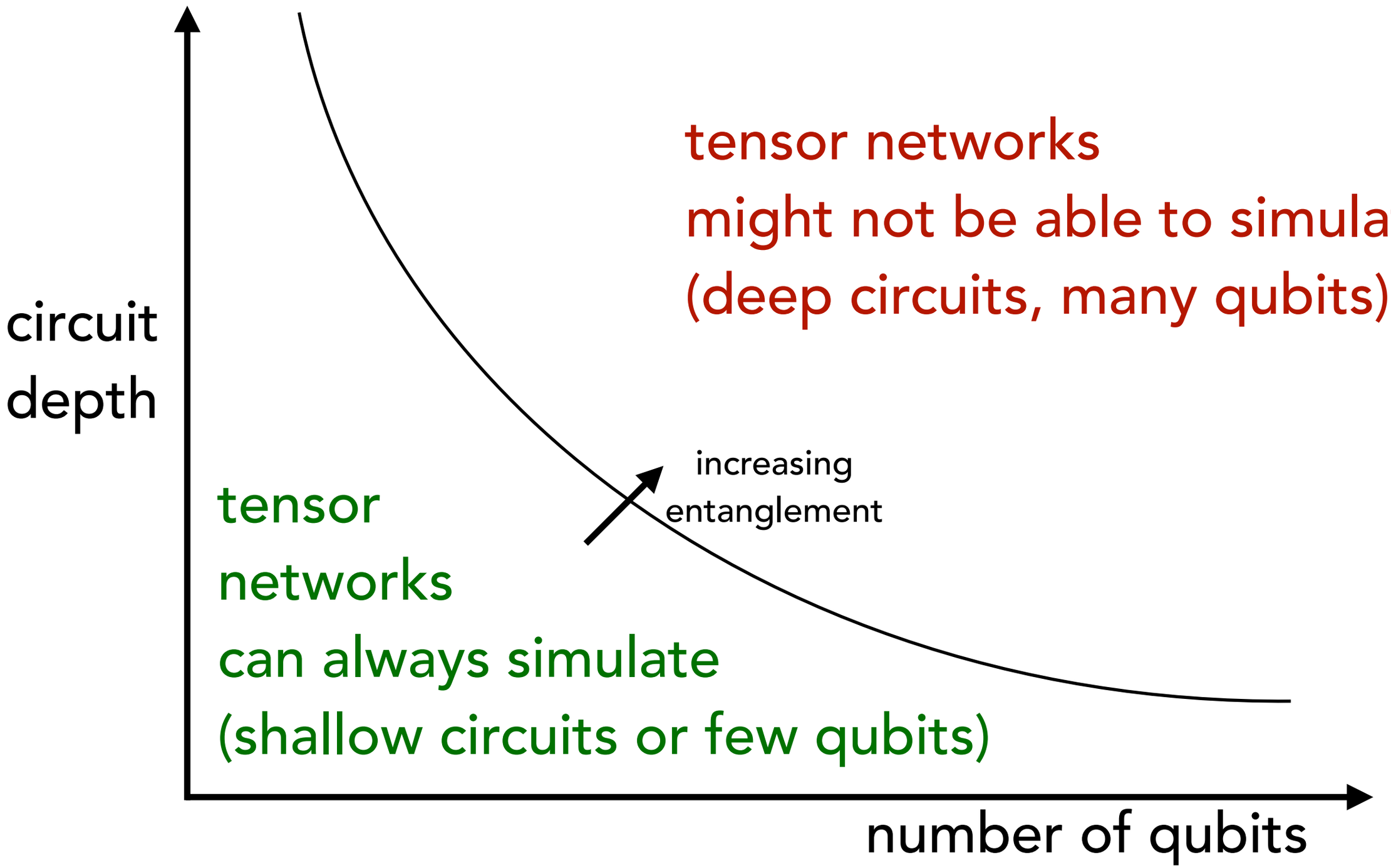
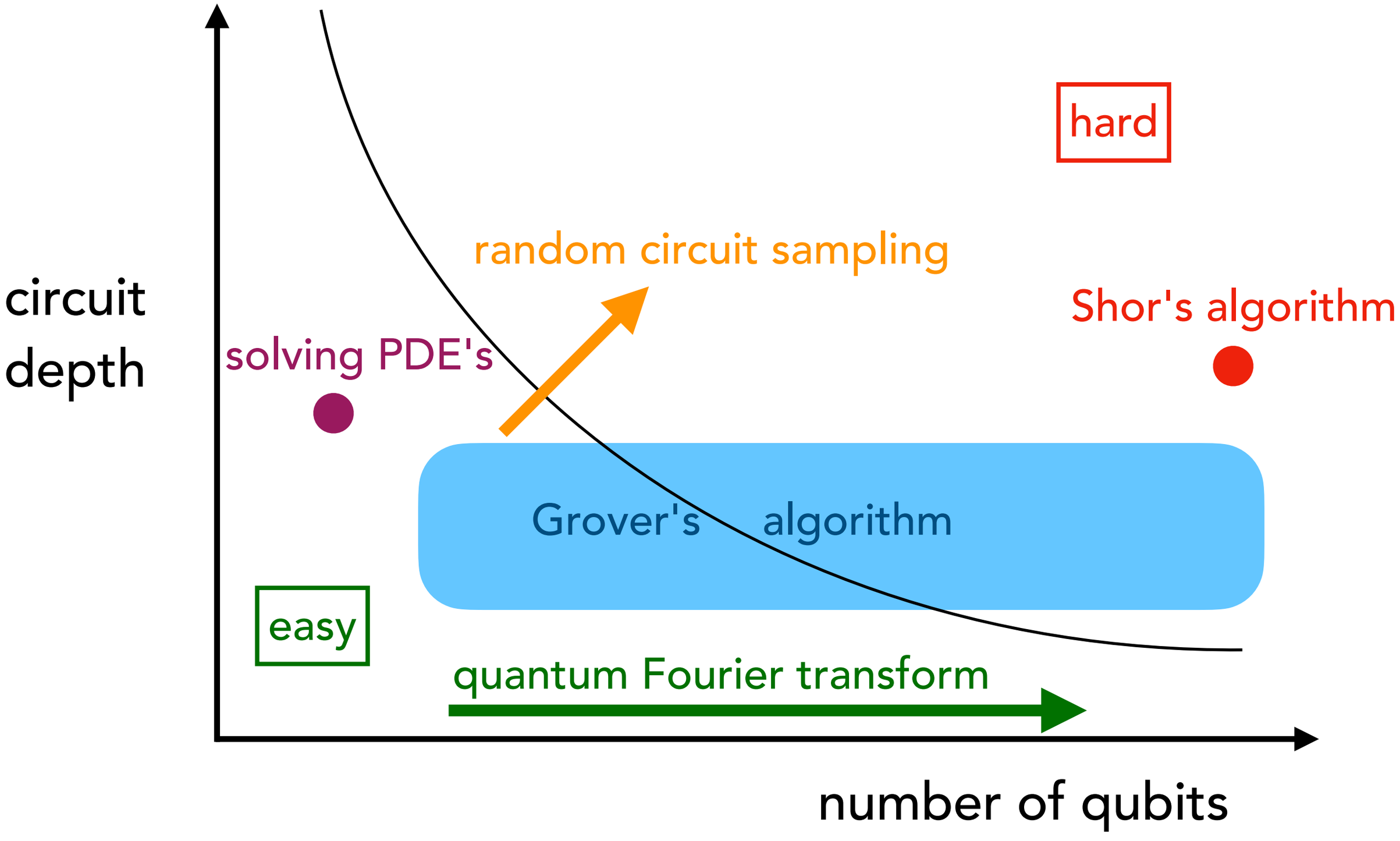


Figure credit: Matt Fishman

*Always true in 1D. Slightly more complicated for 2D +

Tensor Network Simulations

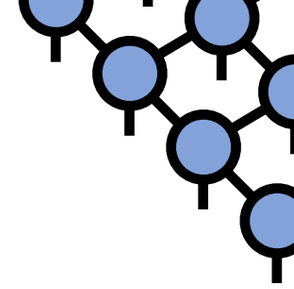
Emerging picture of what is easy vs. hard for tensors



Known capabilities of quantum computer vs quantum-inspired tensor network

	Quantum Computer	Tensor Network
unitary operations	yes	yes
sampling / measurement	yes	yes
non-unitary operations	no	yes
"forced" measurement	no	yes
use unnormalized states	no	yes
moderate entanglement	yes	yes
high entanglement	yes, but how high?	no

Summary & Outlook



Tensor networks can simulate quantum circuits & support quantum-inspired algorithms

Delineates boundary between classical and quantum computing

Growing applications to many areas:

- novel machine learning algorithms
- black-box, gradient-free optimization
- pure mathematics (quantum groups, category theory)
- high-energy physics / QCD